VAX/VMS
Systems Dispatch
September 1985
AD-L034A-27

digital
software

VAX/VMS SYSTEMS DISPATCH

Published by
Corporate Administrative Services Group, Software Services
Digital Equipment Corporation
129 Parker Street (PKO2/E49)
Maynard, MA 01754

The **VAX/VMS Systems Dispatch** contains new and revised Software Product Descriptions, programming notes, software problems statements and responses. Much of the material is developed from Software Performance Report (SPR) answers significant to the general audience and is printed here to supplement the maintenance updates.

DECnet-VAX	VAX CORAL-66	VAX SPM
DECtype/VMS	VAX DATATRIEVE	VAX TDMS
VAX ReGIS Graphics Library	VAX DBMS	VAX 2780/3780 Protocol
RTEM-11	VAX DECalc	Emulator
VAX Ada	VAX DECOR	VAX 3271 Protocol
VAX ADE	VAX DIBOL	Emulator
VAX BASIC	VAX DSM	FORTRAN IV/VAX to RSX
VAX BLISS	VAX FMS	(Cross Compiler)
VAX C	VAX FORTRAN	VAX SORT/MERGE
VAX CDD	VAX MUX200	VAX
VAX COBOL	VAX PASCAL	PDP DATATRIEVE/
	VAX PL/I	VAX

DISTRIBUTION

The **VAX/VMS Systems Dispatch** is directed to one software contact for each software product. No mailing will be made to addresses without a software contact name. **Address change requests should be sent to the nearest DIGITAL field office. Include the new address and mailing label from the most recently received publication.**

Software binary and sources are provided under licenses only. The standard Terms and Conditions, OEM Agreement, and/or Quantity Discount Agreement contain the licenses for all binaries other than DECsystem-10.

Barbara Scollan, Associate Editor

TRADEMARKS of DIGITAL EQUIPMENT CORPORATION
Maynard, Massachusetts

DEC	DIBOL	RSX
DECUS	EDUsystem	RT
digital	IAS	ULTRIX
DECnet	MASSBUS	UNIBUS
DECsystem-10	PDP	VAX
DECSYSTEM-20	PDT	VMS
DECwriter	RSTS	VT

VAX/VMS Systems Dispatch, September 1985

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

HELPFUL HINTS
FOR WRITING
SPRs

HINTS FOR WRITING SPRS

1.0 Introduction

Software Performance Reports (SPRs) exist to benefit customers as well as DIGITAL. They provide informaton to customers and feedback to DIGITAL about software problems.

The following descriptions provide guidelines for submitting information to DIGITAL so that SPR problems can be solved. Some information is common to all SPRs; other information is requested for only certain types of problems.

2.0 SPR Priority Levels

The following explanations of SPR priorities should be used as a guideline for determining the priority of an SPR. Please note that the priority determination should be based on the system or facility behavior that has actually been experienced at the site and should not be based on the perception of the impact of a potential problem.

Priority	Explanation

---

1. MOST PRODUCTION WORK CANNOT BE RUN e.g., important production software is unusable, the system will not boot, necessary peripherals cannot be used as intended, no workaround exists.

2. SOME PRODUCTION WORK CANNOT BE RUN e.g., certain functions or jobs are not usable, level of performance is not as expected or some documented feature does not work as expected but there is a workaround.

3. ALL PRODUCTION WORK CAN BE RUN WITH SOME IMPACT ON USER e.g., significant manual intervention is required, performance has degraded but work can still be done.

4. ALL PRODUCTION WORK CAN BE RUN WITH NO SIGNIFICANT IMPACT ON USER e.g., problem can be patched easily, simple bypass procedure exists.

5. NO SYSTEM MODIFICATIONS NEEDED TO RETURN TO NORMAL PRODUCTION e.g., suggestion, consultation, documentation error or inquiry.

3.0 General Guidelines

This section covers the information that should be provided with all SPRs. Depending upon the problem, this information will vary in quantity and content. Remember that the more pertinent information that is included, the easier it is for DIGITAL to resolve the problem.

3.1 Scenario

A complete scenario should be supplied, usually in the form of a batch log console listing or SET HOST/LOG output file that shows exactly how the problem is produced. Supplying only the output produced by the problem is not enough. The entire scenario of what was done by the user is needed. The problem may be caused by an interaction between various system events, software packages, devices, SYSGEN parameters, DCL symbols or logical names. Some or all of the displays generated by the following commands may be required for different problems:

```
$ SHOW LOGICAL/ALL/FULL
$ SHOW SYMBOL/ALL/GLOBAL

$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>   USE ACTIVE
SYSGEN>   SHOW/ALL
SYSGEN>   SHOW/SPECIAL
SYSGEN>   EXIT
```

3.2 Limit Problem Scope

As much as possible, eliminate all extraneous elements from the scenario. For example, if the execution of a very large program causes a problem, shorten the program to include only the code that causes the problem or write a small program that demonstrates the problem. This action has two benefits: first, logic errors may be discovered; second, the maintainer looking into the problem does not have to comprehend unnecessary material.

3.3 Machine-readable Files

If possible, supply any software needed to reproduce the problem. This may include source programs, image files, sample data, command procedures, logical names etc. If source programs are submitted, also include any libraries or require files referenced. These files must be provided in machine-readable format. Console medium or ANSI magtape are the best media to include with the SPR.

If the problem involves a system crash, include the system dump.

The data should be written to an ODS-2 format disk or an ANSI magtape. For example, the following commands will copy the system dump file to an ANSI magtape:

```
$ INIT MTA0: DUMPS
$ MOUNT/FOREIGN MTA0:
$ BACKUP/IGNORE=NOBACKUP SYS$SYSTEM:SYSDUMP.DMP -
_$ MTA0:DUMPS/SAVE
$ DISMOUNT MTA0:
```

NOTE

Since the system dump file is frequently marked
NOBACKUP (telling the BACKUP utility to copy the
file attributes but <u>not</u> its contents), the dump
file be <u>must</u> copied with:

BACKUP/IGNORE=NOBACKUP

This will insure that the file contents are
copied, as well as the file attributes. The
commands used to write the media should also be
provided with the SPR.

On a MicroVAX, where there is no console block storage device, use one of the
floppy diskette drives to create machine-readable medium to be included with
the SPR. The following commands can be used to copy files:

```
$ INIT  $FLOPPY1: SPRDATA
$ MOUNT  $FLOPPY1: SPRDATA
$ CREATE/DIRECTORY  $FLOPPY1:[DUMP]
$ BACKUP MYDATA.DAT,MYIMAGE.EXE  $FLOPPY1:[DUMP]SPRDATA/SAVE
$ DISMOUNT  $FLOPPY1:
```

On a full VAX, where there is a console block storage device, the following
commands can be used to copy machine-readable data:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  CONNECT CONSOLE
SYSGEN>  EXIT
```

(At this time, remove the console medium and place a scratch volume in the
console block storage device.)

```
$ INIT CSA1: SPRDATA
$ MOUNT CSA1: SPRDATA
$ CREATE/DIRECTORY CSA1:[DUMP]
$ BACKUP MYDATA.DAT,MYIMAGE.EXE CSA1:[DUMP]SPRDATA/SAVE
$ DISMOUNT CSA1:
```

It is important to use BACKUP to write the media submitted with an SPR.
Transferring files in a save set produced by BACKUP is much more reliable than
copying files to the media.

When machine-readable data is not provided in BACKUP save-set format, include
the exact commands that were used to write the data and the commands used for
reading it. Other formats are discouraged, since they may cause problems.

All machine-readable media submitted with SPRs will be returned to the customer.

3.4 System Environment

Every computer site runs a different type of workload. Some problems only appear under certain conditions. For example, some sites give different classes of users different base priorities. These sites may encounter problems that other sites do not. This information can be extremely important in resolving the problem, especially for system hangs or system crashes.

Describe any special software packages that are being used. Also, mention any foreign hardware devices or user-written drivers.

Software version numbers should be included. For example, if there is a problem with accessing local symbols during a DEBUG session, the version numbers of DEBUG and all relevant compilers/assemblers should be specified.

If any patches other than those from maintenance updates are being used, they should be mentioned in the SPR.

3.5 User Analysis (Optional)

Optionally, an analysis of the problem may be included. Any useful miscellaneous information should be included, such as, "Without xyz happening, the problem could not be reproduced" or "On version Vx.y, this problem does not occur."

4.0 Problem-specific Information to Include

Resolution of different classes of problems generally requires different kinds of additional information.

NOTE

> For those items that are identified with a single asterisk (*), the raw data file (SYS$ERRLOG:ERRLOG.SYS), not the formatted output from the ANALYZE/ERROR utility, should be included. Formatted output frequently does not include all the information needed to resolve the problem.
>
> For those items that are identified with a double asterisk (**), the raw data file (SYS$SYSTEM:SYSDUMP.DMP), not the formatted output from the SDA utility, should be included. Formatted output usually does not include all the information needed to resolve the problem.

Problem	Information to Include
System Bugcheck/Crash	A machine-readable copy of the system dump file must be included.** (Output from the SDA utility should not be sent since it usually does not include enough information to resolve the problem). A copy of the error log at the time of the error should also be included because many system problems are triggered by hardware errors.*
Machine-check:	On a machine check, include a machine-readable copy of the error log, not output from the error log generator.* A machine-readable copy of the system dump file should also be included. **
System Hang:	When a system appears "hung" (no response on any terminals), the system should be manually crashed and the system dump file included with the SPR. When the system is shut down in this way, the console listing is very important and should be included with the SPR.

On VAX-11/730, VAX-11/780, VAX-11/782, VAX-11/785, and VAX 8600 primary console terminals, enter: (do nothing on the attached processor's console)

```
^P
HALT
@CRASH
```

7

On VAX-11/750 console terminal,
enter:

```
^P
E P
E/I 0
E
E
E
E
D/G F FFFFFFFF
D P 1F0000
C
```

On MicroVAX I:

Push the HALT button on the front
panel of the CPU box twice, so that
the button is latched out (the red
light in the center of the button
is out).

Then, on the console terminal, enter:

```
E P
E/I 0
E +
E +
E +
E +
D/G F FFFFFFFF
D P 1F0000
C (Then wait a minute or so)
```

Note: If a CRT is being used, copy
the displayed values from the examine
commands to paper and submit them with
the SPR.

The preceding command sequences cause
the VAX or MicroVAX system to bugcheck in a
manner that is recognized by VMS developers
as a forced crash.

Also include a description of the currently
running workload.

8

VAXclusters:

> If all machines in a VAXcluster are "hung" for a reason other than an explainable lack of quorum, a coordinated set of dumps plus console listings from all machines may be required for diagnosis. A coordinated set of dumps is a dump from every machine in the cluster taken in a way that ensures that the lock and other data structures are consistent between all dumps. To take a coordinated dump, first halt every VAX in the cluster. The last machine must be halted no more than 99 seconds after the first machine is halted. After all machines have been halted, crash each machine as directed under SYSTEM HANG, and provide all of the dumps and all of the console logs with your SPR.

Executive:

> If it appears that there is a problem with the executive code, include the active values of the system parameters. These can be obtained by invoking SYSGEN and entering both the SHOW/ALL and SHOW/SPECIAL commands.

> A machine-readable copy of the source program showing the problem plus libraries, require files, and build files should also be included, if possible.

> Also include a copy of the machine-readable error log at the time of the problem. *

Devices:

> For any suspected device or device driver error, include a copy of the error log at the time of the problem. *

Corrupted RMS Files:

> When an RMS file becomes corrupted by software, an SPR should always be submitted. Items to include with the SPR are:

1) A copy of the corrupted file.

2) Any programs (preferably with sources) and data that are necessary to reproduce the corruption. Note the distinction between programs that merely demonstrate that the file is corrupt, as opposed to a program that causes the corruption to occur. Please try to trim down the program to isolate the specific operations that led to the corruption.

3) A description of how the file is being processed when the corruption occurs. For example, how many users are accessing the file, what kind of operations are being performed on the file ($UPDATEs, $PUTs, $DELETEs, etc.).

Sometimes accessing a corrupted file can cause nonfatal bugchecks. If it appears that a process is continually disappearing from the system, check the error log for nonfatal bugchecks. If this is the case, include a crash dump with the SPR. To obtain a crash dump (assuming the system manager has given permission), perform the procedure below. Since this procedure will crash the system, it is suggested that it be performed during off-peak hours. Be sure to give adequate warning if there are any users on the system.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>   USE ACTIVE
SYSGEN>   SET BUGCHECKFATAL 1
SYSGEN>   WRITE ACTIVE
SYSGEN>   EXIT
$ RUN PROGRAM_THAT_BUGCHECKS
```

Intermittent: For a problem that is intermittent or that is not reproducible, include a copy of the machine-readable error log at the time of the problem. *

Command Language
Interpreters: When submitting an SPR on a command
                                language interpreter, it is important
                                to show all symbols and logical names
                                defined on the system by using the
                                following commands:

                                SHOW SYMBOL/ALL/GLOBAL
                                SHOW SYMBOL/ALL/LOCAL
                                SHOW LOGICAL/ALL/FULL

                                Also, indicate whether private or
                                modified command tables are being used.

Job Controller: If the job controller process
                                encounters a fatal error condition,
                                it aborts execution and restarts
                                itself (as a new process). Upon
                                restart, the system job queue file
                                is not reopened automatically;
                                a START/QUEUE/MANAGER command and
                                appropriate START/QUEUE commands
                                must be manually issued to restart
                                batch and print processing for that
                                node.

                                For this type of controller problem,
                                include a copy of the console log
                                error message and a machine-readable
                                copy of the job controller process
                                dump written by the system to
                                SYS$SYSTEM:JOBCTL.DMP. In addition,
                                if the START/QUEUE/MANAGER command
                                fails because of a corrupted system job
                                queue file, also include a machine-
                                readable copy of the queue file.
                                The default queue file name is
                                SYS$SYSTEM:JBCSYSQUE.DAT.

Print Symbiont:

                 Print symbiont process dump:

                                If the print symbiont exits, a message
                                from the job controller is printed
                                on the console, together with an error
                                message from the print symbiont. Also,
                                a symbiont process dump is written

11

to SYS$SYSTEM:PRTSMB.DMP. Include a
copy of these console log messages and
a machine-readable copy of the symbiont
process dump. Also include copies of
the displays:
>   SHOW QUEUE/FULL/ALL
>   SHOW PRINTER (for all
>   printer execution queues)
>   SHOW QUEUE/FORM/FULL
>   SHOW TERMINAL (all terminal
>   execution queues)
If a file was involved, include a
DIRECTORY/FULL of the file and, if
possible, a machine-readable copy of the
file. If at all possible, attempt to
explain the conditions which directly
preceded the symbiont exit, such as
commands used or attempted, and/or a
detailed description of the symbiont
behavior prior to exiting.

Unexpected format or output generated with print
symbiont:

If the print symbiont problem exists in
the formatting or output of data, include
a machine-readable copy of the file and
the library modules in use when printing.

Include a DIRECTORY/FULL display of the
file and a copy of the displays using
the following commands:
>   SHOW QUEUE/FULL/ALL
>   SHOW QUEUE/FORM/FULL
>   SHOW PRINTER and/or SHOW TERMINAL
>   (whichever is applicable)
Along with a description of the explicit
PRINT command, include qualifiers and a
copy of the FILE TRAILER page. Please
provide all information required to repro-
duce the behavior consistently.

User-written and user-modified symbiont problems:

Describe the problem as completely as
possible, including the intent of the user
symbiont. Supply all details surrounding
the problem and include a well-commented
listing of the user-supplied symbiont or

routine. If the problem is associated with the specification of the queue, form, characteristics, parameters, or other input to the DCL command line, include a log file or a description of the PRINT command which demonstrates the problem.

LIBRARIAN:

If there is a problem with the LIBRARIAN, include the following material:

1. A machine-readable copy of the library itself
2. Machine-readable copies of all input files to the library
3. Information (DIRECTORY/FULL) on the library file
4. Information (LIBRARY/LIST/FULL) on the library contents

If the problem can be duplicated at will, include the scenario and any command files used.

LINKER:

If there is a problem with the LINKER, include machine-readable copies of the object files, shareable images, and libraries used in the link, along with a full map (LINK/MAP/FULL).

Debugger:

Include sources, objects, and images for the program being debugged. If the program is large, it would be very helpful to reduce the size of the program to demonstrate the same problem. Also include a log of the debugging session and include the DEBUG.LOG file that the debugger produces.

DECnet:

For a DECnet problem, supply configurations of the systems involved in the problem. This information should include the version numbers of the operating systems and DECnet, the hardware on both systems, and the patch level of the DECnet software on the non-VMS system, if applicable. Depending on the nature of the problem, it might also be applicable to supply hard-copy display of executor, line or circuit parameters and/or counters.

Terminals:

If there is a problem with the terminal driver, provide the following information:

1. A list of terminal characteristics (SHOW TERMINAL)
2. The type of terminal
3. The type of modem (if any)
4. Any special front-end equipment
5. Any unusual terminal configuration

If the problem involves remote file access, it is often useful for the maintainer to know if the same or similar operation can be performed from a different account, or with the source and destination nodes reversed.

Compiler/Assembler:

If there is a problem with the assembler or a compiler, include the source program that caused the problem. (It is very important to include all require files and libraries that are referenced by the source program).

It is especially important to limit the scope of the problem when submitting SPRs on compilers.

Include the version number of the compiler and the version number of the operating system.

NEWS BULLETIN

Important VAX/VMS Version 4.2 Information

The following information helps clarify the installation procedures contained in the VAX/VMS Release Notes, Version 4.2, and stresses the importance of installing or upgrading correctly.

There are two methods by which VAX/VMS Version 4.2 can be brought up.

METHOD 1

Perform a complete installation onto a blank disk. This method has no restrictions. If a system is currently running VAX/VMS Version 3.n, there might be problems with the installation of layered products.

METHOD 2

Perform an upgrade to an existing system. This method has two restrictions:

1. The system to be upgraded must be VAX/VMS Version 4.0 or later. The upgrade will not work on any earlier version of VAX/VMS.

2. The upgrade kit and files must be on a device that is distinct from the system device. Most upgrades are done directly from the distribution media. Do not copy the files from the distribution media to the system disk. The upgrade does not work under these circumstances.

In either case, the mandatory update must be applied after the upgrade or installation has finished. In the case of MicroVMS, the update must be applied after the base kit, and again after any options or layered products have been installed. The installation of MicroVMS options and layered products will not succeed unless the mandatory update has been applied.

17

KNOWN PROBLEMS AND RESPONSES

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 5.20.8
          PRODUCT: VAX/VMS
        COMPONENT: SYS

Lack of disk quota causes ERRFMT to fail

PROBLEM
STATEMENT

After upgrading from VAX/VMS Version 3.7 to VAX/VMS Version 4.0,
the ERRFMT process fails to run. This failure is a result of the
ERRFMT process having neither disk quota on the system disk nor
the EXQUOTA privilege.   Prior to VAX/VMS Version 4.0, the ERRFMT
process was given the EXQUOTA privilege.

RESPONSE

The ERRFMT process is no longer given the EXQUOTA privilege. This
avoids the potential problem of a malfunctioning device consuming
excessive   space   on   the   system   disk   with   entries.   This
modification creates an unintentional side effect of disabling
error logging on systems which have disk quotas enabled on the
system device, but no entries for the ERRFMT's UIC.

In a future update to VAX/VMS, we intend to document this behavior
more completely and, if necessary, allocate a reasonable quantity
of disk quota for the ERRFMT process.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 5.20.9
         PRODUCT: VAX/VMS
       COMPONENT: SYS

GETJPI ("","TERMINAL") truncates names

PROBLEM
STATEMENT

When using VAX/VMS Version 4.0 with virtual terminals enabled,
after many logins, the virtual terminal names reach a length of
over 7 characters ("VTA1000:").

These terminal names, when returned by $GETJPI, are truncated to
7 characters ("VTA1000:" is truncated to "VTA1000"), producing an
error for the following case:

```
$ TERM = F$GETJPI("","TERMINAL")
$ DEFINE/USER SYS$INPUT 'TERM'
$ EDIT FOO.COM
```


RESPONSE

It is true that $GETJPI returns an invalid terminal name if the
name length reaches 7 characters (the colon is dropped from the
returned string even though it returns an 8 byte string). We
expect that this problem will be fixed in a future major release
of VAX/VMS.

Note, however, that excluding the missing colon, the actual device
name is correct. Therefore, the following workaround can be used,
regardless of the length of the terminal name:

```
$ TERM = F$GETJPI("","TERMINAL")
$ TERM = TERM - ":" + ":"
$ DEFINE/USER SYS$INPUT 'TERM'
$ EDIT FOO.TXT
```

The second command insures that a colon is appended to the end of the terminal name whether or not one already exists.

This problem exists for any call to $GETJPI with the item code JPI$_TERMINAL. Programs which call $GETJPI should include code to handle the case of a missing colon.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 10.15.1
         PRODUCT: VAX/VMS
       COMPONENT: STARTUP

Terminal logical names in UVSTARTUP.COM

PROBLEM
STATEMENT

If a MicroVAX contains more than one terminal controller, the
MicroVMS logical names $TERMINAL1, $TERMINAL2, etc., might be
defined incorrectly.

RESPONSE

In a future update, we expect to correct UVSTARTUP.COM so it does
not try to define these logical names twice if there is more than
one terminal controller.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 11.15.1
         PRODUCT: VAX/VMS
       COMPONENT: LOGINOUT

Incorrect validation of MAXJOBS

PROBLEM
STATEMENT

Batch process login is incorrectly validated against the user
quota MAXJOBS. If MAXJOBS is set to N, N+1 jobs are admitted.

RESPONSE

This problem is corrected in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 11.15.2
        PRODUCT: VAX/VMS
      COMPONENT: LOGINOUT

DEFCLI prohibits CLI table change in SPAWN

PROBLEM
STATEMENT

The DEFCLI flag prohibits the user from changing command tables in
a SPAWN command. No error message is given; rather, the spawned
process executes with the standard command tables, giving
potentially unexpected results.

RESPONSE

Since the CLI tables are viewed as an extension of the CLI, it is
necessary to restrict changing them to enforce the intent of the
DEFCLI flag correctly.

In a future update of VAX/VMS, we expect to give a suitable error
message when SPAWN/CLI is specified.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 11.15.3
          PRODUCT: VAX/VMS
        COMPONENT: LOGINOUT

Network jobs not counted against MAXJOBS

PROBLEM
STATEMENT

The authorization record fields MAXJOBS and MAXACCTJOBS do not
apply to network processes. As a result, a user performing remote
file access operations can consume system resources beyond normal
limits.

RESPONSE

The exemption of network jobs from the job quotas results from the
behavior of network file access. It is possible for the accessor
to close a file and request access to a new file before the
NETSERVER process (running FAL) is ready to accept the next
transaction.   Under this circumstance, DECnet starts a new
NETSERVER process. As a result, it is possible to create multiple
NETSERVER processes on the remote node, all serving the same
series of requests.   Denying the creation of any of these
processes causes the file access operation to fail at the
initiating node. The most common occurrence of this situation is
when the user performs a wildcard remote file access operation.

In VAX/VMS Version 4.2, network processes are counted against the
MAXJOBS quota with an exemption of up to four network processes
per user. This represents a compromise between preventing users
from flooding the system and allowing wildcard network file access
to function correctly.

At some later date, we expect to provide a more comprehensive
solution to this problem.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 11.30.1
          PRODUCT: VAX/VMS
        COMPONENT: SYSBOOT

TOPSYS system root is incorrect

PROBLEM
STATEMENT

VMS does not handle the VMB input, TOPSYS, correctly. Somewhere
between the time when VMB finds SYSBOOT and when the STARTUP.COM
command procedure is executed, the value of TOPSYS is altered from
[SYS5.] to [SYS0.].

RESPONSE

The behavior described above can be caused by booting VMS using a
pre-Version 4 copy of VMB. While it does not seem likely that an
old version of VMB would exist on the Version 4 system disk, it is
possible that an old version exists on the console media. Older
versions of VMB do not pass the TOPSYS input through to the
running system, and a default of [SYS0.] is used.

Because VMB is decoupled from the rest of the VMS system, it is
our goal that VMB be forward and backward compatible with the rest
of VMS. Care is taken to insure that new versions of VMB will
boot the previous version of VMS and that old versions of VMB will
boot the next version of the VMS operating system. However,
forward compatibility is restricted by the fact that old versions
of VMB will never be able to boot using hardware or software
capabilities that are introduced in the newer versions of VMS.

A workaround is to update the VMB on the console media with the
VMB.EXE from the VAX/VMS Version 4.0 system disk.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 11.40.2
        PRODUCT: VAX/VMS
      COMPONENT: SYSINIT

SYSUAF.DAT redefined for bypass at login

PROBLEM
STATEMENT

In the MicroVMS User's Manual, page 1-16, under Section 1.2.5.1,
"Alternate User Authorization File" there are instructions for
breaking into a locked system. This method does not work. If the
instructions are followed and "ANYTHING" is typed for the user
name and password, an error occurs indicating user authorization
failure.   There is no alternate UAF file present on the system.
If SET UAFALTERNATE 1 is set at the SYSBOOT prompt, the correct
system password (that is, the one in the default UAF file) still
allows access to the system.

RESPONSE

The   procedure   does   not   work   as   documented   because of   an
incorrectly defined logical name in SYS$SYSTEM:SYSINIT.EXE. The
MicroVMS User's Manual, Chapter 1, covers the correct procedure
for using an alternate user authorization file to break into a
locked system.

The error in SYSINIT.EXE is corrected in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0
       PRODUCT: VAX/VMS
   COMPONENT: VMSINSTAL

Seq. 12.15.2

VMSINSTAL GET option fails on Version 4 update

PROBLEM
STATEMENT

The GET option on VMSINSTAL does not work for the Version 4.0 mandatory update because the BACKUP save sets do not follow expected naming conventions.

RESPONSE

This problem will be avoided by using proper save set naming conventions for future updates of VAX/VMS.

A workaround is to use COPY to preserve all necessary BACKUP save sets.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 15.30.3
           PRODUCT: VAX/VMS
         COMPONENT: PRINT SYMBIONT

Cannot bypass all formatting in print symbiont

PROBLEM
STATEMENT

The VAX/VMS Version 4.0 standard print symbiont does not provide a
mechanism to suppress all formatting or bypass the buffering of
output. The PRINT/PASSALL command does not allow data to be
passed to the driver without the formatting of output.

RESPONSE

The PRINT command qualifier /PASSALL, supported in VAX/VMS Version
4.0, is intended to allow the user to bypass interpretation of
data by the main format routine.   However, the current
implementation of this qualifier in the standard symbiont does not
perform as intended.

In VAX/VMS Version 4.2, the DCL command PRINT/PASSALL bypasses all
formatting.

In the meantime, formatting by the symbiont may be avoided by
using form definitions and the /PASSALL and /NOFEED qualifiers of
the DCL PRINT command. The insertion of carriage return and line
feed at the right margin may be avoided by defining a form as
/NOWRAP and /NOTRUNCATE.

For example:

        $ DEFINE/FORM /NOWRAP /NOTRUNCATE /STOCK=DEFAULT
        _Form name: NOFORMAT
        _Form number: 2

This form, together with the /PASSALL qualifier, must be specified on the PRINT command. For example, to print the file FOOBAR.TXT, use the following command:

    $ PRINT/FORM=NOFORMAT /PASSALL FOOBAR.TXT

The insertion of form feeds by the symbiont at the bottom margin can be avoided by using the DCL command:

    $ SET QUEUE/DEFAULT=NOFEED

Not all formatting will be bypassed with the above suggestions. However, defining a form /NOWRAP and /NOTRUNCATE and setting the queue /DEFAULT=NOFEED might help.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 15.30.4
       PRODUCT: VAX/VMS
     COMPONENT: PRINT SYMBIONT

Print symbiont allocates output device

PROBLEM
STATEMENT

It is not possible to start an output queue on a device that is
allocated to another user. Issuing the DCL command START/QUEUE
when another process has the device opened for output results in
an error.

RESPONSE

This behavior is intentional. When the output queue is started,
the print symbiont allocates and assigns a channel to the output
device.   The allocation of the device prevents users from
accessing the device. The output device is allocated until the
output queue is stopped.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 15.30.5
        PRODUCT: VAX/VMS
      COMPONENT: PRINT SYMBIONT

Multiple page headers generated by plot

PROBLEM
STATEMENT

PRINT/HEADER/NOFEED is used to produce a plot with a page header
on an LXY22 plotter. The print symbiont prints a header line, a
form feed, and then continues the true plot.   During the
processing of the true plot, more page headers are output by the
symbiont. The page header appears to be output to the device
whenever the total number of lines on the page is exceeded. It
appears the symbiont is counting the number of records written and
issuing page headers when the line count exceeds that of the form
definition.

RESPONSE

The VAX/VMS Version 4.0 print symbiont creates a page header for
output to the device if page headers are active and the current
page is considered a new page. A page is considered new if the
number of lines left on the page is zero. The number of lines
left on the page is equal to the form length minus the current
line count. If the bottom margin is nonzero, the form length is
further reduced by the size of the bottom margin. The current
line count is changed whenever a vertical format effector is
encountered by the standard VAX/VMS Version 4.0 print symbiont.

The prefix carriage control character of every record in a
particular FORTRAN file is a hexadecimal 20 or space character. As
documented in the VAX/VMS I/O User's Reference Manual, Part I, pp.
5-6 through 5-8 and the VAX/VMS Utility Routines Reference Manual,
pp. 5-21 through 5-22, a space character in the prefix area of a
FORTRAN carriage control record indicates that a line feed should
be inserted before the FORTRAN record.

The print symbiont performs the interpretation of FORTRAN carriage control prefix character, inserts the required format effector, and, depending upon the interpretation, changes the current column, line or page position.   Line feed, in this case, increments the current internal line count of the print symbiont. A new page is generated by the symbiont when the current line count of the symbiont exceeds the length of the form. A page header is created if page headers are active and a new page is generated by the symbiont.

A single page header at the top of the page can be achieved by creating a form with a length that exceeds the total number of line feeds specified by the records.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 15.30.6
         PRODUCT: VAX/VMS
       COMPONENT: PRINT SYMBIONT

Loss of print job when carrier is dropped

PROBLEM
STATEMENT

When carrier is dropped on a remote printer, the current print job
is not requeued as specified in the VAX/VMS Release Notes, Version
4.0, but is lost and must be resubmitted.

RESPONSE

The VAX/VMS Release Notes, Version 4.0, incorrectly states that
the current job is requeued for printing upon loss of carrier.
The default behavior for output queues upon detection of error is
loss of the current job. This default behavior can be overridden
by setting the queue attribute for retention of jobs upon
detection of error.

The following DCL command sets this queue attribute for the output
queue named FOO:

     $ SET QUEUE/RETAIN=ERROR FOO

If a job is retained in an output queue as a result of the queue
attribute /RETAIN=ERROR, operator intervention is required to
release that job for printing. The following DCL command releases
a retained entry for printing in the output queue FOO:

     $ SET QUEUE/ENTRY=entry-number /RELEASE FOO

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 15.30.7
        PRODUCT: VAX/VMS
      COMPONENT: PRINT SYMBIONT

File left open by print symbiont

PROBLEM
STATEMENT

When attempting to dismount a user disk, it is discovered that the
print symbiont has left a file open.

The following two conditions exist:

    1. The symbiont is idle with no current jobs.

    2. The queues are not spooled to this disk.

RESPONSE

This problem is fixed in VAX/VMS Version 4.2.

As a temporary workaround, stop all queues served by that symbiont
with the DCL command:

    $ STOP/NEXT queue

This causes the symbiont process to close all files and exit.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 15.30.8
         PRODUCT: VAX/VMS
       COMPONENT: PRINT SYMBIONT

Implicit spooling restricts user

PROBLEM
STATEMENT

Software that creates plots for a non-DIGITAL plotter uses
implicit spooling for the submission of plot files to the output
queue. When the plot files are written by the print symbiont,
form feeds are inserted. A workaround is to create an
intermediate file and to submit this file for plotting using the
DCL command PRINT/NOFEED. Setting the output queue to
/DEFAULT=NOFEED does not work because other users rely on the
form feed capability of the symbiont for this output device.

The carriage control type of the file is embedded (none) carriage
control.

RESPONSE

Implicit spooling is not a recommended method for submitting jobs
to an output queue. Implicit spooling restricts the user to
system defaults. The ability to specify job attributes, such as
pagination (/FEED), is lost. The recommended method for
submitting jobs to output queues under program control is the
VAX/VMS Version 4.0 system service $SNDJBC.

The $SNDJBC system service is documented on pages 322 to 359 in
the VAX/VMS System Services Reference Manual. This service
provides greater control of the symbiont processing of the job.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 15.30.9
         PRODUCT: VAX/VMS
       COMPONENT: PRINT SYMBIONT

Print symbiont performs tab expansion

PROBLEM
STATEMENT

The VAX/VMS Version 4.0 print symbiont does not allow horizontal
tabs to be expanded by the device.

RESPONSE

When the output queue is started, the VAX/VMS Version 4.0 print
symbiont determines if tab expansion is required by accessing the
current device characteristics.   The Version 4.0 print symbiont
expands horizontal tabs only when the device is incapable of
handling the tab character.   On a device controlled by the
LCDRIVER or LPDRIVER, the DCL command SET PRINTER/TAB sets the tab
characteristic for that device.   On a serial line controlled by
the terminal driver, the DCL command SET TERMINAL/TAB sets the tab
characteristic for that serial device.

The device characteristics for a particular output queue are
determined at the start of that output queue.   Therefore, set the
device characteristics before starting the output queue.   If the
characteristics of a device need to be reset after the output
queue has been started, stop the queue, reset the device
characteristics, and then restart the output queue.   The output
queue must be completely stopped before changing any device
characteristics.

We will consider enhancing the print symbiont to access the device
characteristics more frequently.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 15.30.10
        PRODUCT: VAX/VMS
      COMPONENT: PRINT SYMBIONT

Print symbiont process termination

PROBLEM
STATEMENT

The print symbiont exits with unexpected symbiont process termination
and a symbiont process dump. The print symbiont exits on jobs with a
block length of zero. The following error message is printed on the
console log:

      %%%%%%%%%%%% OPCOM   date-time %%%%%%%%%%%%
      Message from JOB_CONTROL
      %JBC-W-SYMDEL, unexpected symbiont process termination

      %%%%%%%%%%%% OPCOM   date-time %%%%%%%%%%%%
      Message from JOB_CONTROL
      -SYSTEM-F-ACCVIO, access violation, reason mask=6E, virtual
                     address=03000000, PC=00000000, PSL=2FFC0000

Deleting, aborting, and requeuing jobs occasionally causes the symbiont
process to abort. The following error message is printed on the
console log:

      %%%%%%%%%%%% OPCOM   date-time %%%%%%%%%%%%
      Message from JOB_CONTROL
      %JBC-W-SYMDEL, unexpected symbiont process termination

      %%%%%%%%%%%% OPCOM   date-time %%%%%%%%%%%%
      Message from JOB_CONTROL
      -PSM-F-BADLOGIC, internal logic error detected at PC 0000B46E

RESPONSE

There are two known problems which result in the unexpected symbiont
process termination. One problem is associated with the creation of
file name banner characters on file separation pages. This problem

46

only occurs when the current job contains no file identification. The second known problem is a result of an improper communication handshake between the job controller and the print symbiont.

Problem in the creation of file separation pages

If LOGINOUT receives an error when attempting to open a batch log file, a special job is sent to the print symbiont. This job contains no file identification. Instead, the job contains the error message returned to the job controller by LOGINOUT. If file separation pages are enabled in the VAX/VMS Version 4.0 print symbiont, and such a job is queued to a running output queue, the symbiont attempts to use the file identification when creating the file separation pages. An access violation results.

This problem is corrected in VAX/VMS Version 4.2. Deleting the offending print job and restarting the output queues controlled by this symbiont process will clear the problem. Rebooting with a new queue file is not necessary. As a temporary workaround, set the block limit on the output queues to print jobs with a block size from 1 to infinity. The following DCL command sets the block from 1 to infinity for the output queue FOO:

    $ SET QUEUE /BLOCK_LIMIT=(1,"""") FOO

Jobs with a block size of zero are retained in the queue. The system manager can either delete these jobs or turn off file separation pages and print the error messages.

Problem in the communication between the job controller and the print symbiont

In VAX/VMS Version 4.0, the print symbiont maintains an internal data structure for each output queue. This internal data structure is reset upon the arrival of each task (file to process). Arrival of tasks is controlled by a communication handshake between the job controller and the print symbiont. When the print symbiont receives a task, it resets the internal data structure, performs some initialization, and then responds to the job controller's start task message. Similarly, deletion of a task requires a response to the job controller's message.

If the symbiont receives a deletion message from the job controller during initialization but before responding to the job controller's start task message, the symbiont improperly responds to the deletion request before responding to the start task request. Internally, the

symbiont continues processing the task until the task can be cleanly aborted. The job controller, on the other hand, schedules any pending task for processing. If the symbiont is still processing the aborted task, it resets its internal data structure upon receipt of the second task. Since this data structure maintains all the state information for the current task, resetting this data structure creates unpredictable results. Subsequently, the symbiont may either abort with a dump or loop internally.

This problem is corrected in VAX/VMS Version 4.2. In the meantime, avoid issuing delete, abort or requeue requests while the print symbiont process is starting, stopping, or aborting a job.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 15.30.11
         PRODUCT: VAX/VMS
       COMPONENT: PRINT SYMBIONT

Print symbiont enters compute loop

PROBLEM
STATEMENT

1.   The print symbiont enters a compute loop after a paused
     output queue is started. The output queue paused near the
     end of the job.

2.   It a queue is in a paused state and the operator issues the
     DCL command START/QUEUE/BACKWARD=n, where n is greater than
     the current page position, the print symbiont process enters
     a compute loop.

RESPONSE

Both of these problems are resolved in VAX/VMS Version 4.2.

Please avoid any attempts to pause an output queue near the end of
the job. Attempting to pause the output queue, when a small print
job is executing or attempting to position at or near the end of
the job, is especially prone to initiating print symbiont process
looping. Also avoid operator commands which may position the
output queue before the first page of the job.

If the print symbiont process enters a compute loop, determine the
print symbiont process identification.   Then issue the DCL
command:

     $ STOP/IDENTIFICATION=symbiont_process_id

Privileges are required to stop the print symbiont process. All
output queues controlled by this symbiont process must be
restarted after deletion of the looping process.

49

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 15.30.12
         PRODUCT: VAX/VMS
       COMPONENT: PRINT SYMBIONT

Miscellaneous problems in print symbiont

PROBLEM
STATEMENT

In VAX/VMS Version 4.0, the following formatting and nuisance
problems have been identified:

1.   There are problems with vertical tab, bolding, and
      underlining when the left margin is nonzero.

2.   When the VAX/VMS Version 4.0 DCL command PRINT /NOFLAG
      /HEADER is issued, the header is not printed on the
      first page of the first copy.

3.   Failure to cancel CTRL/O results in all jobs submitted
      to the controlling queue to be lost until a subsequent
      CTRL/O is issued on the device.

4.   There is improper termination of escape sequences. This
      causes data embedded in an escape sequence to be
      improperly interpreted.

5.   An extra blank line is generated on paper when the
      bottom margin is nonzero and /FEED is enabled.

6.   No flag page is printed when the burst page is
      requested. Multiple rows of burst characters are not
      printed on the perforation between the flag page and
      burst separation pages.

RESPONSE

These problems are corrected in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 15.30.13
        PRODUCT: VAX/VMS
      COMPONENT: PRINT SYMBIONT

Serial printers on DMF disconnect

PROBLEM
STATEMENT

Printers on the serial lines of the DMF-32 often disconnect. This
presents problems on output queues controlling the serial line;
jobs might be lost or corrupted. The terminal line is set /NODMA.

RESPONSE

There is a problem in the DMF-32 driver with regard to device
timeouts. An incorrect algorithm is used to calculate the amount
of time to wait for the I/O to complete. The amount of time
calculated is abnormally small. If the terminal sends an XOFF at
the "right" time, the device might time out. If this terminal is
in use as a printer, then upon timeout, the current job aborts.
Subsequently, the following error message is sent to the output
device by the print symbiont:

    %PSM-E-WRITEERR, error writing TXA0
    -SYSTEM-F-TIMEOUT, device timeout

We expect to fix this problem in a future update of VAX/VMS. As a
temporary workaround, set the terminal /DMA. This avoids using
the incorrect timeout value. The following DCL command sets this
attribute for the terminal TXA0:.

    $ SET TERMINAL TXA0: /DMA

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 20.5.1
        PRODUCT: VAX/VMS
      COMPONENT: DCL

Captive account causes LOGINOUT access violation

PROBLEM
STATEMENT

A captive account, which cannot access its login command file,
causes an access violation in LOGINOUT.

RESPONSE

This problem is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 20.5.2
          PRODUCT: VAX/VMS
        COMPONENT: DCL

Cannot change/examine logical name table prot

PROBLEM
STATEMENT

VAX/VMS Version 4.0 allows protection masks to be specified when
creating a logical name. However, there is no corresponding
capability to change and/or examine the protection assigned to a
table.

RESPONSE

The ability to examine the current protection mask of a logical
name table is provided in VAX/VMS Version 4.2.

The ability to change the protection mask of an existing logical
name table will be considered for implementation in a future
release of VAX/VMS.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 25.5.5
          PRODUCT: VAX/VMS
        COMPONENT: DECnet

STREAM_LF file transfer hangs to non-VMS partners

PROBLEM
STATEMENT

An attempt to access a file having STREAM_LF format using DECnet
sometimes results in the process hanging if the remote node is not
a VAX/VMS system.

RESPONSE

This behavior is caused by a failure in the error recovery logic
within RMS. This logic performs the various message traffic
necessary to recover from a reported error during a file transfer.
The logic contains an error which produces an invalid error
recovery message and typically results in both partners waiting
for an input message.

This problem was corrected in VAX/VMS Version 4.1.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 25.5.6
         PRODUCT: VAX/VMS
       COMPONENT: DECnet

DECnet gives incorrect error on invalid user name

PROBLEM
STATEMENT

If an invalid user name is used in a DECnet access control string,
the error returned is SS$_LINKEXIT instead of SS$_INVLOGIN.

RESPONSE

This problem is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 31.10.1
          PRODUCT: VAX/VMS
        COMPONENT: DDDRIVER

TU58 times out when /DATA_CHECK=WRITE is used

PROBLEM
STATEMENT

When a TU58 cartridge is mounted using the /DATA_CHECK=WRITE
qualifier for the DCL command MOUNT, timeout errors occur on
subsequent write operations if the I/O operation crosses a track
boundary. This occurs when using the CONSCOPY utility to make a
duplicate copy of the console TU58 on a VAX-11/730 or a
VAX-11/750, but can occur under other circumstances as well.

RESPONSE

The root of the problem is that the default timeout value for I/O
operations performed by the TU58 device driver is only long enough
to accommodate one complete traverse of the tape.

The TU58 is a block-addressable tape device, which logically
appears to consist of four tracks of 128 blocks each. When a
track boundary is crossed during a read or write operation, the
tape must be rewound to the beginning before the operation can
continue.

However, when a write data-check is performed, the tape must be
positioned to the beginning of the data which was just written. If
this data straddles a track boundary, a second traverse of the
tape is required.

In a future update of VAX/VMS, we expect to correct this problem
by increasing the timeout value if a data-checked I/O operation is
being performed.

In the meantime, avoid using the /DATA_CHECK=WRITE qualifier.
Note also that the status of the error returned by EXCHANGE during
a CONSCOPY operation is "informational" rather than "fatal"
indicating that EXCHANGE has recovered from the error.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 35.5.1
        PRODUCT: VAX/VMS
      COMPONENT: EDIT/ACL

EDIT/ACL deletes ACE granting access

PROBLEM
STATEMENT

If a user has gained control access to a file via the access
control list (ACL), using the ACL editor to modify the ACL
results in a privilege error at the end of the editing session.
In addition, the ACL for the file is deleted.

RESPONSE

This problem is a result of the way the ACL editor writes the
modified ACL back to the file. First, the entire ACL is deleted
so that the new version can be written. However, if access to the
file is gained via an access control list entry (ACE), the ACE
granting access is deleted. When the attempt is made to write out
the new ACL, the user does not have the necessary privileges
because the ACE has been deleted.

We expect this problem to be corrected in a future update of
VAX/VMS.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 35.5.2
         PRODUCT: VAX/VMS
       COMPONENT: EDIT/ACL

Problem in refresh logic causes access violation

PROBLEM
STATEMENT

After doing about six successive refresh (Gold + CTRL/R)
operations, the ACL editor fails with an access violation.

RESPONSE

This problem is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 35.5.3
        PRODUCT: VAX/VMS
      COMPONENT: EDIT/ACL

Missing status return

PROBLEM
STATEMENT

If the user is able to read a file's access control list (ACL),
but does not have the necessary privileges to change it, the ACL
editor will exit with a success status when ending an editing
session.

RESPONSE

This problem is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.5.3
        PRODUCT: VAX/VMS
      COMPONENT: CONVERT

CONVERT incorrectly returns RTL error

PROBLEM
STATEMENT

When converting input files with fixed-length records, CONVERT
might report the following error, even though the records are the
correct length:

    %CONVERT-I-RTL, record longer than maximum length

RESPONSE

This behavior occurs because CONVERT computes unnecessarily large
buffer sizes.

We expect that this problem will be fixed in a future update of
VAX/VMS. As a workaround, specify the /TRUNCATE qualifier on the
CONVERT command. Note that the addition of this qualifier does
not affect the validity or structure of the output file, but
allows CONVERT to process the file successfully.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.6
        PRODUCT: VAX/VMS
      COMPONENT: RMS

File corruption with global buffers

PROBLEM
STATEMENT

Periodically, RMS bucket format check failures occur on a number
of shared files.

RESPONSE

This problem is fixed in VAX/VMS Version 4.2. In the meantime,
disabling global buffers from the file should provide a temporary
workaround.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.7
        PRODUCT: VAX/VMS
      COMPONENT: RMS

SYS$RMSRUNDWN returns incorrect status

PROBLEM
STATEMENT

SYS$RMSRUNDWN fails to return the status of RMS$_NORMAL on
successful completion of the service. The documentation states
that SYS$RMSRUNDWN should be repeatedly called until the value
RMS$_NORMAL is returned.

RESPONSE

The success path out of SYS$RMSRUNDWN corrupts the value of R0.
At the moment, it is impossible to receive this return value; the
call returns an alternate success code of either SS$_WASSET or
SS$_WASCLR.

This behavior is corrected in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.8
          PRODUCT: VAX/VMS
        COMPONENT: RMS

Search list questions

PROBLEM
STATEMENT

Unexpected behavior is observed with search lists. First, the DCL
command   CREATE/DIRECTORY   creates   directories   in   the   last
translation of a search list. Second, RMS produces RMS-E-DNF
errors when it attempts to find a nonexistent file in a search
list.

RESPONSE

The first problem, concerning the behavior of CREATE/DIRECTORY,
was corrected in VAX/VMS Version 4.1. It creates the directories
in the first translation of a search list.

The second behavior is correct. This becomes more apparent when
considering that a search list specifically defines a set of
places. RMS assumes that if a specific set of places is defined,
they all exist; it cannot do otherwise. The RMS-E-DNF errors are
returned because RMS is told to look in a set of directories, some
of which do not exist.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.9
         PRODUCT: VAX/VMS
       COMPONENT: RMS

Remote command procedures fail

PROBLEM
STATEMENT

Execution of a remote command procedure fails if it invokes an
image.   As an example, given a command procedure, B.COM,
consisting of the lines:

    $ RUN FOO
    $ DIRECTORY
    $ EXIT

the DCL command:

    @A::B

fails with the error:

    %DCL-W-SKPDAT, image data (records not beginning with "$")
    ignored.

RESPONSE

VAX/VMS Version 4.0 RMS contains an error which causes it to fail
to detect the end of image data in a remote command procedure. As
a result, whenever a remote command procedure invokes an image,
RMS proceeds to read the entire command file before reporting an
end-of-file condition to DCL. This causes DCL to report
erroneously that image data is skipped. Also, DCL does not
attempt to execute any further lines in the command procedure.

This problem was corrected in VAX/VMS Version 4.1.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 40.45.10
         PRODUCT: VAX/VMS
       COMPONENT: RMS

Version 4 COPY will not copy Version 3 ISAM files

PROBLEM
STATEMENT

Attempts to copy certain RMS ISAM files created on a VAX/VMS
Version 3 system to a VAX/VMS Version 4 system fail with the
following message:

    %COPY-W-INCOMPAT, input and output have incompatible
    attributes

RESPONSE

This problem occurs as a result of the interaction of RMS and
COPY. COPY objects to the input file because the maximum bucket
size in the file header is larger than any of the buckets actually
in the file. VAX/VMS Version 3 RMS allowed ISAM files to be
created this way.

VAX/VMS Version 4 RMS creates the output file with a maximum
bucket size that truly reflects the maximum bucket size of any
area in the file, and COPY notices the difference.

At this time, there are no plans to modify this behavior. As a
workaround, use CONVERT from the VAX/VMS Version 4 system to
produce a file with a valid maximum bucket size in the file
header:

    $ CONVERT V3SYS::disk:[dir]file.dat *

Alternatively, use EDIT/FDL to set the BUCKET_SIZE secondary under
the FILE primary to be the maximum of all of the BUCKET_SIZE
values under the AREA n primaries.

A subsequent CONVERT/FDL of the file on the VAX/VMS Version 3 system yields a valid file that can then be transferred successfully using COPY. See the VAX/VMS File Definition Language Facility Reference Manual for more details on using EDIT/FDL, as well as further information on the structure of FDL files.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.11
        PRODUCT: VAX/VMS
      COMPONENT: RMS

RMS file parse problem with level 8 directories

PROBLEM
STATEMENT

The following command sequence fails on VAX/VMS Version 4.0:

        $ SET DEFAULT DISK1:[A.B.C.D.E.F.G.H]
        $ DIRECTORY SYS$DISK:[]
        %RMS-F-DIR, error in directory name

RESPONSE

When RMS attempts to parse a file name that contains an empty
directory element, it temporarily includes this null directory as
a directory level in the file name. Since this occurs after the
default directory string has been applied, default directories
that already contain eight elements fail with RMS-F-DIR.

We expect to correct this behavior in a future update of VAX/VMS.
Until then, this problem is easily avoided by not specifying the
null directory string in the file names and allowing RMS to fill
this field in as the final step to file name processing.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 55.10.3
         PRODUCT: VAX/VMS
       COMPONENT: AUTHORIZE

Clarification of ADD/NETWORK

PROBLEM
STATEMENT

When attempting to specify access limits on users by using
AUTHORIZE, the following command denies all access:

    UAF> ADD JOE /NETWORK/NOINTERACTIVE/NOBATCH

Is it possible to insure that only NETWORK access is granted?

RESPONSE

The above command is incorrectly parsed by AUTHORIZE, and all
access is denied.

In VAX/VMS Version 4.2, this behavior is fixed so that the command
executes properly and allows only NETWORK access.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 55.10.4
         PRODUCT: VAX/VMS
       COMPONENT: AUTHORIZE

AUTHORIZE and DISKQUOTA do not return status

PROBLEM
STATEMENT

For ease of command procedure use, utilities such as AUTHORIZE and
DISKQUOTA should be enhanced to support foreign command line
interfaces; they should return the status of each invocation in
the symbol $STATUS.

RESPONSE

AUTHORIZE presently has a foreign command interface. The correct
status is returned in VAX/VMS Version 4.2.

DISKQUOTA does not presently have a foreign command interface,
but we will consider adding one which returns status for a future
release of VAX/VMS.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 55.10.5
          PRODUCT: VAX/VMS
        COMPONENT: AUTHORIZE

Problem with SHOW/ID followed by MOD/ID

PROBLEM
STATEMENT

The MODIFY/IDENTIFIER command fails when preceded by the following
SHOW/IDENTIFIER command.

```
UAF> modify/identifier GUSTAV_MAHLER /attribute=resource
%UAF-I-RDBMDFYMSG, identifier GUSTAV_MAHLER modified
UAF> show/identifier GUSTAV_MAHLER
    Name                        Value               Attributes
    GUSTAV_MAHLER               %X80010029          RESOURCE
UAF> modify/identifier GUSTAV_MAHLER /attribute=resource
%UAF-E-RDBMDFYERR, unable to modify identifier GUSTAV_MAHLER
-SYSTEM-F-IVIDENT, invalid identifier format
```

Also, the MODIFY/IDENTIFIER/HOLDER command fails in all contexts.

```
UAF> modify/identifier/holder=JOE GUSTAV_MAHLER /attribute=resource
%UAF-E-RDBMDFYERR, unable to modify identifier GUSTAV_MAHLER
-SYSTEM-F-IVIDENT, invalid identifier format
```


RESPONSE

Improper coding in the MODIFY/IDENTIFIER logic causes the working
parameters to be set up incorrectly under these circumstances.

Both of these problems are corrected in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 55.50.1
        PRODUCT: VAX/VMS
      COMPONENT: DEBUG

SET MODULE command takes too long

PROBLEM
STATEMENT

The DEBUG command:

    DEBUG> SET MODULE V40SHR

where V40SHR is an 18000 block shareable image module supplied by
the user, takes over 2 hours elapsed time to complete.

RESPONSE

When DEBUG executes the SET MODULE command, it reads the global
symbol table in the shareable image and builds an internal table
of symbols that is sorted by address. The time required for the
Version 4.0 sorting algorithm is proportional to $N**2$, where N is
the number of symbols in the module.

In VAX/VMS Version 4.2, the sorting code uses an algorithm which
is proportional to $N*(\log N)$. As a result, it takes approximately
20 seconds of CPU time to perform the same SET MODULE command.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 55.50.2
         PRODUCT: VAX/VMS
       COMPONENT: DEBUG

Comma lists on DEPOSIT not allowed

PROBLEM
STATEMENT

Under VAX/VMS Version 3.n, a user could perform the following
command, which would deposit the specified values into successive
locations:

    DBG> DEPOSIT 200=1,2,3

Under VAX/VMS Version 4, the above command produces a syntax error
response.   The documentation implies this feature should still
work. The HELP text does not show this syntax.

RESPONSE

Large portions of VAX DEBUG have been rewritten for VAX/VMS
Version 4.   The current implementation does not allow successive
deposits   from   a   single   DEPOSIT   command.   However,   the
documentation for the DEPOSIT command has not been changed to
reflect the new behavior.

In a future revision, the documentation of the DEPOSIT command
will be corrected to reflect the current behavior.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 56.20.3
       PRODUCT: VAX/VMS
     COMPONENT: LINKER

Version 4.0 images larger than Version 3.0 images

PROBLEM
STATEMENT

The size of image files linked on VAX/VMS Version 4.0 can be
substantially larger than those linked before VAX/VMS Version 4.0.

RESPONSE

When the linker produces an image that has several consecutive
uninitialized pages, it creates an image section descriptor in the
image header that describes these pages, which are called
demand-zero pages. This prevents the uninitialized pages from
occupying disk space and results in a smaller image file. When
the image is later executed, the image activator interprets this
image section descriptor and marks these pages as demand-zero in
the process' address space.

However, these additional demand-zero image sections incur
additional image activator overhead, since they must be treated
individually. This is usually not a problem, but images with a
large number of images sections incur excessive image activator
overhead when run. Because of trade-offs such as these, the
linker provides a number of parameters which can be set by the
user, but have defaults that are usually adequate.

When the linker surpasses 96 (default value) image sections during
the production of an image, it puts a priority on minimizing the
number of additional image sections. This is done by disabling
the demand-zero optimization mentioned above. Note that this can
result in a larger image file, since each uninitialized image
page has a corresponding disk block.

This is the intended behavior, but it is easily controlled. The
parameter ISD_MAX (see the VAX/VMS Linker Reference Manual, page
LINK-24) controls the upper limit on image section creation before
demand-zero compression is disabled. The number of image sections
in an image is listed on the last page of the linker map. Set
ISD_MAX to some value greater than the number of image sections.

The occurrence of this problem is made more likely by the breakup
of the VAX/VMS Run-Time Library (VMSRTL), especially if the image
file contained almost 96 image sections before VAX/VMS Version
4.0.   If the image file references more than one of the
individual shareable images that now comprise VMSRTL, more image
sections are added to the image file than were added before
VAX/VMS Version 4.0.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 56.30.1
        PRODUCT: VAX/VMS
      COMPONENT: MAIL

MAIL cannot run on a GIGI terminal

PROBLEM
STATEMENT

MAIL will not activate on a GIGI terminal and no error messages
are produced.

MAIL uses the SMG screen management package, which does not
support GIGI terminals.

RESPONSE

In VAX/VMS Version 4.2, MAIL displays an error message for
undefined terminals.

In addition, SMG includes GIGI support in VAX/VMS Version 4.2.

Thus, starting with VAX/VMS Version 4.2, it will be possible to
run MAIL on a GIGI terminal.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 56.40.1
        PRODUCT: VAX/VMS
      COMPONENT: MONITOR

Foreign terminal support does not work

PROBLEM
STATEMENT

MONITOR works improperly on foreign terminals.

RESPONSE

The MONITOR utility shipped with VAX/VMS Version 4.0 uses the RTL
screen package (SCR$...) which is documented in the VAX/VMS
Release Notes, Version 4.0 under the "Obsolete RTL Routines."
Foreign terminals are supported with this screen package through
the use of the SYS$EXAMPLES:SCRFT.MAR program. This program must
be modified to handle the needs of a foreign terminal. All of the
information necessary for doing this is contained within
SCRFT.MAR.

Because of a problem with the RTL screen package's handling of
virtual memory, the foreign terminal support does not work in
VAX/VMS Versions 4.0 and 4.1.

This problem is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0
          PRODUCT: VAX/VMS   Seq. 56.40.2
        COMPONENT: MONITOR

MONITOR's virtual memory usage grows continuously

PROBLEM
STATEMENT

MONITOR's virtual memory size continuously increases. If a process
runs MONITOR for a sufficient length of time, it eventually
terminates with an insufficient virtual memory error.

RESPONSE

This problem is caused by the screen management package which
MONITOR uses for display. Under certain circumstances, the screen
management package does not free up the same amount of virtual
memory which it allocated. In this case, the amount freed is less
than the amount allocated, which causes the continuous growth in
virtual memory usage and eventually the insufficient virtual
memory error.

This problem is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 56.52.1
         PRODUCT: VAX/VMS
       COMPONENT: PURGE

PURGE can incorrectly delete files

PROBLEM
STATEMENT

If a file specification appears twice in a PURGE command line, all
versions of the file are deleted. For example, when a user issues
the following command, all versions of FOO.BAR are deleted:

    $ PURGE FOO.BAR,FOO.BAR

RESPONSE

This problem is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0 Seq. 56.80.2
PRODUCT: VAX/VMS
COMPONENT: SET

Volume retention dates override SET FILE dates

PROBLEM
STATEMENT

Under VAX/VMS Version 3, it was possible to reset the expiration
date of a file to any date, without regard to the date set by
having volume retention enabled. This is no longer the case with
VAX/VMS Version 4.0.

RESPONSE

We expect this problem to be corrected in a future update of
VAX/VMS.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 56.80.3
        PRODUCT: VAX/VMS
      COMPONENT: SET PASSWORD

SET PASSWORD always returns success status

PROBLEM
STATEMENT

The DCL command SET PASSWORD returns SS$_NORMAL in $STATUS,
regardless of the success or failure of the command.

RESPONSE

This behavior is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 56.85.1
       PRODUCT: VAX/VMS
     COMPONENT: SHOW CLUSTER

CNX_STATE documentation error

PROBLEM
STATEMENT

The SHOW CLUSTER utility displays the value NEW for the CNX_STATE
of the MEMBERS class. The description for NEW states:

> The system has just booted and the state of the
> connection is too new to be determined.

No other value for this field has been seen on the local CPU.

RESPONSE

The status value of NEW for the connection state field, CNX_STATE,
as described in the VAX/VMS Show Cluster Utility Reference Manual,
page SHCL-27, is incorrect.

We expect to correct this documentation error in a future release
of VAX/VMS. The correct description for the value NEW should be:

> No attempt to make a connection has been made yet.

NEW is the correct status value for the local node.   The
Connection Manager does not attempt to form a VAXcluster
connection to itself on the local node.

Note that the description of the NEW value for the CNX_STATE field
is correct in SHOW CLUSTER's online HELP. HELP for the CNX_STATE
field can be obtained by invoking SHOW CLUSTER with the
/CONTINUOUS qualifier and typing HELP FIELD CNX_STATE.

89

OPERATING SYSTEM: VAX/VMS V4.0
        PRODUCT: VAX/VMS
      COMPONENT: SHUTDOWN

SHUTDOWN$INFORM_NODES usage described

PROBLEM
STATEMENT

When the orderly system shutdown procedure, SYS$SYSTEM:SHUTDOWN.COM, is
run, all users running on all VAXcluster members receive notification
of the system shutdown. This might cause users to log out, assuming
that the node they are using is shutting down. However, this is often
not the case.

The orderly system shutdown procedure appears to have code to notify
users of a system shutdown using the REPLY /NODE= command, but it does
not work correctly.

RESPONSE

The REPLY /NODE= option in the orderly system shutdown is activated by
defining the process logical name SHUTDOWN$INFORM_NODES before invoking
the shutdown procedure. For example, the command sequence:

        $ DEFINE SHUTDOWN$INFORM_NODES MOE,LARRY
        $ @SYS$SYSTEM:SHUTDOWN

will cause only the users on VAXcluster members MOE and LARRY to be
informed of the system shutdown. We recognize that this feature is not
described in any VAX/VMS documentation, and we will attempt to remedy
that oversight as soon as possible.

The primary considerations for the default orderly system shutdown
behavior are twofold. First, a user logged into any VAXcluster member
might be affected by the shutting down of any other VAXcluster member.
The user might have batch jobs running on the other VAXcluster member
or, if terminal servers are in use, the user might have an alternate
terminal session active on the VAXcluster member that is being shut
down.

Therefore, we believe all users on all VAXcluster members must be informed of the shutdown of any VAXcluster member. Second, all orderly shutdown messages include the name of the VAXcluster member being shut down. A user need only check the shutdown message carefully to avoid logging out of a system unnecessarily.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 56.90.2
          PRODUCT: VAX/VMS
        COMPONENT: SHUTDOWN

Time-of-year clock causes SHUTDOWN error

PROBLEM
STATEMENT

Occasionally, during an orderly system shutdown, the SYS$SYSTEM:SHUTDOWN
command procedure exits quickly with the following error messages:

        %SET-E-NOTSET, error modifying time
        -SYSTEM-F-IVTIME, invalid time

However, a SHOW TIME command performed immediately after the error occurs
always shows the correct time. The problem can be corrected by performing
one or more SET TIME commands specifying the current time.

RESPONSE

We believe this problem results from an improperly functioning
time-of-year hardware clock. VMS maintains the time of day using two
clocks. While the system is running, a memory quadword is updated at
10-millisecond intervals. This is the quadword current system time
discussed in the descriptions of VMS time-related system services.   In
addition, most VAX processors provide a time-of-year clock in hardware,
which maintains the current time when the processor is not operating. The
time-of-year clock allows VMS to "know" the correct time whenever it is
booted. When no time parameter is specified, as is the case during an
orderly system shutdown, the SET TIME command recalibrates the memory
quadword time using the time-of-year hardware clock.

The "invalid time" error is produced by the recalibrate function. This
happens whenever the recalibrate function determines that the quadword
current system time is more than 1 day ahead of the time given by the
time-of-year clock. Executing a SET TIME command that specifies the
current time before performing the orderly system shutdown eliminates the
error; this causes VMS to reset the time-of-year clock.

An error might occur if the time-of-year clock has lost a day or more since the last time the system was booted or the last time a SET TIME command was executed, whichever occurred more recently.

In addition, to eliminate the errors mentioned above, we are considering changing the way that the orderly system shutdown command procedure uses the SET TIME command.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 57.10.1
          PRODUCT: VAX/VMS
        COMPONENT: SPAWN

Cannot specify spooled device with SPAWN

PROBLEM
STATEMENT

Spawning by using either the DCL SPAWN command or the LIB$SPAWN
routine fails when the explicitly specified output stream is a
spooled device.

RESPONSE

This problem is fixed in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 57.10.2
         PRODUCT: VAX/VMS
       COMPONENT: SPAWN

LIB$SPAWN fails with MBFULL

PROBLEM
STATEMENT

When LIB$SPAWN is invoked with RESOURCE_WAIT disabled, on a
heavily loaded system, it returns an error status:

    %SYSTEM-W-MBFULL, mailbox is full

When the system is lightly loaded, the call to LIB$SPAWN is
successful. Increasing the system default mailbox size also
rectifies the error.

RESPONSE

While not documented in the VAX/VMS Run-Time Library Routines
Reference Manual under LIB$SPAWN, the VAX/VMS DCL Dictionary
states on page DCL-745 that one of the restrictions governing a
SPAWN operation is the requirement that RESOURCE_WAIT mode be
enabled for the process. This is also documented in the VAX/VMS
System Messages and Recovery Procedures Reference Manual on page
2-295, where it discusses the MBFULL error message.

To insure that LIB$SPAWN does not incur the MBFULL error, enable
RESOURCE_WAIT for the process via the SET PROCESS/RESOURCE_WAIT
command. This synchronizes the mailbox I/O between the parent
process and the subprocess. Increasing the system default mailbox
size only masks the problems, allowing a greater system load
before exhibiting the MBFULL error condition.

OPERATING SYSTEM: VAX/VMS V4.1
        PRODUCT: VAX/VMS
      COMPONENT: DOCUMENTATION

SYS$GETJPI documentation errors

PROBLEM
STATEMENT

Page SYS-198 of the <u>VAX/VMS System Services Reference Manual</u>
indicates that the symbols JPI$_OTHER, JPI$_NETWORK, JPI$_BATCH,
and JPI$_INTERACTIVE are defined in the library module $JPIDEF.
These symbols are documented incorrectly; they should all have a
"K" inserted in their names (for example, JPI$K_OTHER,
JPI$K_NETWORK, etc.).

RESPONSE

The four symbols that appear in the table at the top of page
SYS-198 should have a "K" in their names. This is to distinguish
these symbols, return values for the JPI$_MODE item, from item
codes whose names are of the form JPI$_item-code.

These errors are corrected in the documentation for VAX/VMS
Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 65.5.5
        PRODUCT: VAX/VMS
      COMPONENT: DOCUMENTATION

SYS$GETJPI documentation error

PROBLEM
STATEMENT

The JPI$_IMAGNAME item code of the $GETJPI system service is
incorrectly documented.

When the JPI$_IMAGNAME item code is specified, $GETJPI does not
return an 8-byte character-string descriptor that points to the
name of the current image file. Instead, it returns the name
itself in the buffer specified in the item list.

RESPONSE

This item code description is corrected in the documentation for
VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 65.5.6
         PRODUCT: VAX/VMS
       COMPONENT: DOCUMENTATION

CHAN argument incorrect for $GETDVI

PROBLEM
STATEMENT

The description of the CHAN argument for $GETDVI is incorrect. The
CHAN argument of the $GETDVI service is passed by value instead of
by reference as stated in the VAX/VMS System Services Reference
Manual.

RESPONSE

The correction will be reflected in a future revision of the
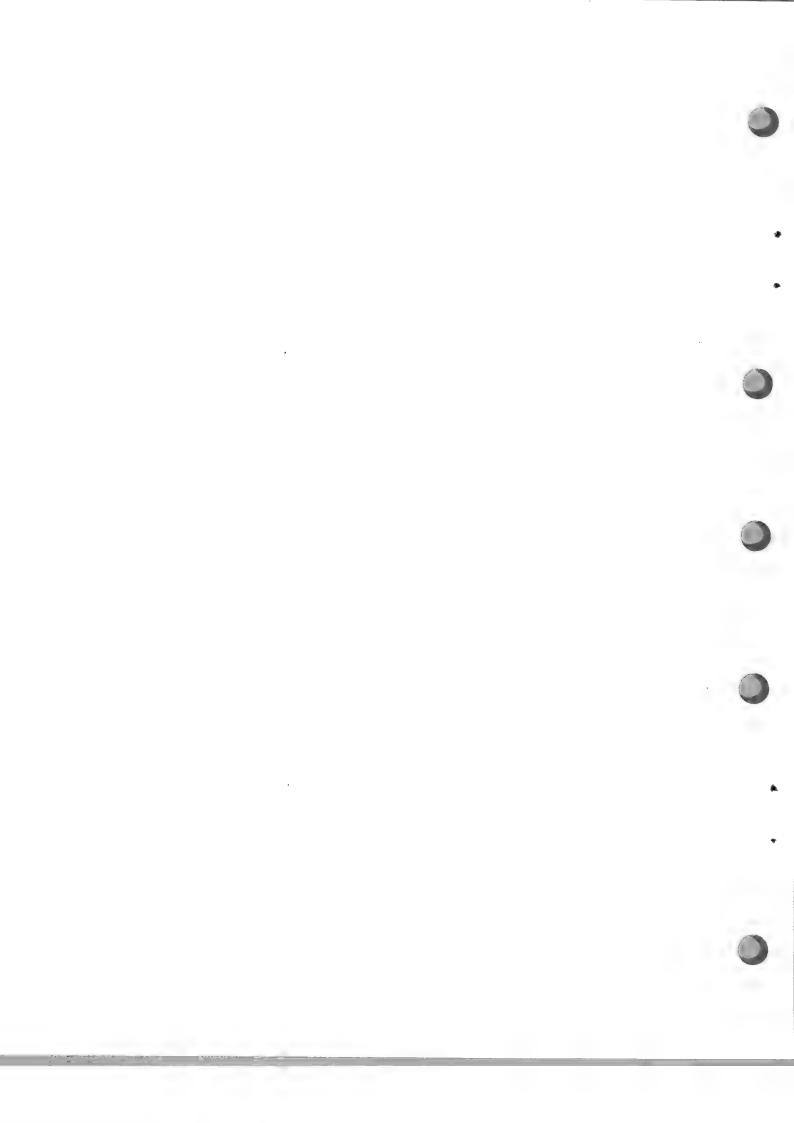documentation.

REPUBLISHED ARTICLES

REPUBLISHED ARTICLES

The following articles are being republished to correct inaccuracies. Five articles for RMS and two articles for CONVERT have been resequenced to reflect the new components numbering system. The technical information in these articles remains unchanged, however, there are minor editorial changes.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.5.1
        PRODUCT: VAX/VMS
      COMPONENT: CONVERT

CONVERT/RECLAIM may access violate

PROBLEM
STATEMENT

When attempting to perform a CONVERT/RECLAIM on an RMS ISAM file,
an access violation occurs. The file is large, yet very sparsely
populated as a result of many record deletions.

RESPONSE

The index structure of the file is corrupt. However, we believe
that CONVERT/RECLAIM should be able to determine that the data it
is processing is incorrect. We expect to enhance CONVERT/RECLAIM
to be more robust in a future update of VAX/VMS.

A file corrupted in this manner can be recovered by issuing a
CONVERT command such as:

    $ CONVERT OLD.DAT NEW.DAT

This will produce a fresh version of the file, although it will be
considerably smaller than the original file because of the
scarcity of records in OLD.DAT. In order to preallocate space for
the new file to allow for the eventual insertion of more records,
use an FDL file which describes the eventual size of the file.
EDIT/FDL provides an easy method to specify these parameters.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.5.2
        PRODUCT: VAX/VMS
      COMPONENT: CONVERT

CONVERT can incorrectly report DUP and SEQ errors

PROBLEM
STATEMENT

CONVERT mistakenly complains about segmented keys being out
of order when creating a Prolog 1 or 2 ISAM file.

RESPONSE

CONVERT compares segment-by-segment, even if an earlier
segment showed the current key to be greater than the
previous key. Since all key segments are extracted and
concatenated for Prolog 3 files, this problem can be avoided
by using Prolog 3 files.

We expect to correct this problem in a future update of
VAX/VMS.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.1
         PRODUCT: VAX/VMS
       COMPONENT: RMS

Read from SYS$OUTPUT fails

PROBLEM
STATEMENT

With VAX/VMS Version 4.0, the user cannot read from SYS$OUTPUT. A
FORTRAN program, such as the following:

    READ (6,*) VALUE
    END

fails with the error message:

    %FOR-F-ERRDURREA, error during read
    unit 6 file SYS$OUTPUT:.;
    user PC 00000412
    -RMS-F-FAC, record operation not permitted by
    specified file access (FAC)

RESPONSE

VAX/VMS Version 4.0 only allows writes to SYS$OUTPUT (instead of
reads and writes, as in previous versions of VAX/VMS) because it
is enforcing that the stream is meant for output. Previous
versions of VAX/VMS incorrectly allowed reads from SYS$OUTPUT (and
writes to SYS$INPUT).

However, since many existing programs depend on this capability,
the old behavior is restored in VAX/VMS Version 4.2.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.2
         PRODUCT: VAX/VMS
       COMPONENT: RMS

COPY/OVERLAY fails if destination write-protected

PROBLEM
STATEMENT

COPY/OVERLAY does not work if the destination file is located on a
remote node and the specified version number already exists. Instead
of copying the data from the source file to the destination, COPY
produces the following error message:

    "%COPY-E-OPENOUT, error opening NODE::FILE.EXT;1 as output"
    "-RMS-E-FEX, file already exists, not superseded"

RESPONSE

If a file is protected against write access by the network accessor, an
access conflict occurs. If a file is available for write access, the
COPY/OVERLAY operation succeeds with no errors. If the destination is
protected against the write access request, the operation fails, since
the COPY utility uses the create-if option when the /OVERLAY qualifier
is specified. For DECnet file operations, this option is simulated
within RMS as an $OPEN or "on-error" $CREATE sequence. This simulation
is attempting the $CREATE under circumstances other than the definition
of create-if allowed. For instance, if the destination file is locked
against write access, RMS incorrectly attempts to create a new file.
Since the version in question already exists, the $CREATE operation
then fails with the error:

    "-RMS-E-FEX, file already exists, not superseded"

In VAX/VMS Version 4.2, RMS returns the correct error from the $OPEN
and does not attempt to create a new file.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.3
         PRODUCT: VAX/VMS
       COMPONENT: RMS

Confusion on $CREATE using search lists

PROBLEM
STATEMENT

An attempted file creation fails when the destination directory is not
located using the first translation of a search list.

Attempting, for example, to create:

    SYS$SYSROOT:[SYSMGR.MGRUTIL]FOO.BAR

fails if SYS$SYSROOT is defined as a search list of 1DUA21:[SYS7.],
SYS$COMMON:, where SYS$COMMON is 1DUA21:[SYS7.SYSCOMMON.] and the
destination directory is SYS$COMMON:[SYSMGR.MGRUTIL].

RESPONSE

The RMS $CREATE service does not attempt to explore a search list
fully. Instead, it only attempts to create new files using the first
translation of any search list logical name that it encounters. In the
example,   the   first   translation   of   the   destination
(1DUA21:[SYS7.SYSMGR.MGRUTIL]) did not exist.

This behavior is documented in the VAX Record Management Services
Reference Manual on page RMS-40.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.4
         PRODUCT: VAX/VMS
       COMPONENT: RMS

RENAME returns incorrect error message

PROBLEM
STATEMENT

For VAX/VMS Version 4.0, the RMS $RENAME service returns the
following error when the file to be renamed does not exist:

    %RMS-E-ACC, ACP file access failed

Prior to VAX/VMS Version 4.0, $RENAME returned the following
error, which was much more meaningful and appropriate:

    %RMS-E-FNF, file not found

RESPONSE

This problem was fixed in VAX/VMS Version 4.1.

OPERATING SYSTEM: VAX/VMS V4.0   Seq. 40.45.5
         PRODUCT: VAX/VMS
       COMPONENT: RMS

Access control string parsed incorrectly

PROBLEM
STATEMENT

In addition to the user name and password, if an account name
is specified in the access control string of a DECnet node
specification, there are cases when the password is
incorrectly conveyed to the remote node. For example, this
occurs when attempting to copy a file from a remote node to
the local system.

RESPONSE

Several utilities, including COPY, use the RMS option
open-by-file-id for accessing their input files.   This
function is simulated for DECnet file access, and the logic
within RMS which performs this does not properly detect a
masked password in the access control string. Thus, the
masked password is not replaced by the real one.

This problem is fixed in VAX/VMS Version 4.2.

VERSION 4 ENHANCEMENTS

VERSION 4 ENHANCEMENTS

We are introducing a new section in the Dispatch entitled "Version 4 Enhancements." During the development of VAX/VMS Version 4, VMS added many new features, enhanced numerous components, and changed and modified existing code. In future issues of the Dispatch, VMS developers will be writing articles clarifying these changes that, based on problem reports we have received, appear to be sources of confusion.

Enhancements in Version 4.0 DCL

The following list provides an abstract of major VAX/VMS Version 3.n DCL problems and suggested enhancements that have been addressed in VAX/VMS Version 4.0.

PROBLEMS

PROBLEM

> The SPAWN and ATTACH commands function improperly when executed from a command procedure.

RESPONSE

> The use of SPAWN and ATTACH from within a command procedure is now fully supported.

* * *

PROBLEM

> Executing a command procedure having STREAM_LF format causes the process to abort.

RESPONSE

> Command procedures written in STREAM_LF format now function properly.

* * *

117

PROBLEM

The LIB$SPAWN RTL routine tranlates input/output file specifications in both the parent process and subprocess if the file specification contains a concealed process logical name. This requires the presence of the process logical name table in the subprocess.

RESPONSE

The VAX/VMS Version 4.0 behavior parses the input/output file specifications completely prior to propagating them to the subprocess.

* * *

PROBLEM

There is currently no way to propagate the current state of the parent's command tables to a subprocess in a SPAWN operation. Therefore, any new commands which are added and are not present in the default command tables, are not inherited by the subprocess.

RESPONSE

The /TABLES qualifier on the SPAWN command allows an installed private command table to be used by a subprocess.

* * *

PROBLEM

DCL does not trap symbols that are greater than 512 bytes. The symbol value given by the SHOW SYMBOL command is truncated without any notification.

RESPONSE

The SHOW SYMBOL command now displays an optional informational message indicating symbol value truncation.

SUGGESTIONS

SUGGESTION

The CLI$DCL_PARSE utility routine should be enhanced to support the use of continuation lines and prompt for missing parameters.

RESPONSE

CLI$DCL_PARSE now includes both of these features.

* * *

SUGGESTION

A DCL command should exist that shows the version limits, set by SET DIRECTORY/VERSION_LIMIT or SET FILE/VERSION_LIMIT commands, for a given file or group of files.

RESPONSE

The DIRECTORY/FULL command now shows version limits currently in effect. This is displayed in the file attributes section.

* * *

SUGGESTION

The F$LOGICAL lexical function should allow the logical name table, used in the translation, to be returned and/or specified.

RESPONSE

A new lexical function, F$TRNLNM, is present in VAX/VMS Version 4.0 which replaces F$LOGICAL. F$LOGICAL is now obsolete, although it still exists for compatibility reasons. One of the many new features of F$TRNLNM is the ability to specify the logical name table to be searched.

SUGGESTION

> The ability should exist to specify a timeout period for the INQUIRE command and perform some default action whenever a timeout occurs.

RESPONSE

> The READ/TIME_OUT command provides this ability.
>
> The DCL command READ/PROMPT="string"/TIME_OUT=n performs essentially the same function as the INQUIRE command. If the READ command terminates with a timeout, it sets the $STATUS value to %X181B0 (RMS-W-TMO). The $STATUS symbol can then be used to branch conditionally to an error routine.

* * *

SUGGESTION

> It should be possible to redirect all of DCL's output to a file.

RESPONSE

> This is possible by creating a supervisor-mode (default mode for the ASSIGN and DEFINE commands) logical name in the process table which equates SYS$OUTPUT to a file name. This automatically creates and opens the file. Deassign SYS$OUTPUT to close the file and redirect the output back to the default device.

* * *

SUGGESTION

> It should be possible to retrieve the name of the command procedure currently executing.

RESPONSE

> This ability exists in the form of a new lexical function, F$ENVIRONMENT. Specify F$ENVIRONMENT("PROCEDURE") to return the name of the currently executing command procedure.

ARTICLES OF GENERAL INTEREST

OPERATING SYSTEM: VAX/VMS V4.1   Seq. 95.5.5
        PRODUCT: VAX/VMS
      COMPONENT: DUDRIVER

System disk mount verification timeout

PROBLEM
STATEMENT

A cluster hang followed an HSC failure. Both systems hung;
sometime later, one system crashed and the other continued to
hang.

RESPONSE

The sequence of events that resulted in this behavior appears to
be as follows.

The HSC experienced a problem which eventually resulted in an HSC
crash and reboot. Both CPUs went into a hang condition since they
were both in mount verification, waiting for the system disk
(which was on the HSC).   One CPU eventually timed out mount
verification on the system disk, and any further I/O to the system
disk failed with a page read error because the volume was invalid.
This is the reason the system crashed. The other CPU continued to
hang because it had not yet timed out the mount verification or
simply had no I/O outstanding on that disk at the time.
Eventually, it might have experienced a crash like the first CPU.

A mount verification timeout usually results in the volume being
invalid until it can be dismounted and remounted. Since it is not
possible or meaningful to dismount the system disk, mount
verification timeout on the system disk is fatal.

We believe that the error which caused the HSC to crash and reboot
is directly related to either hardware or microcode within the
HSC. If this problem continues, field service should examine all
the HSC console logs to determine the problem.

If mount verification timeout on the system disk continues to be a problem, consider raising the value of the SYSGEN parameter MVTIMEOUT. This increased value should equal the time which the user is willing to wait for a disk before giving up and attempting to continue without it.

We expect to modify mount verification in a future VAX/VMS update so that the system disk never experiences mount verification timeouts.

OPERATING SYSTEM: VAX/VMS   Seq. 95.5.6
          PRODUCT: VAX/VMS
        COMPONENT: RMS

DIRECTORY and search list confusion

PROBLEM
STATEMENT

Using a common system disk, if the following command sequence is
performed:

    $ SET DEFAULT SYS$MANAGER
    $ DIRECTORY SYS$COMMON:SYSTARTUP.COM

two copies of the file are displayed even though there is only one
file present. However, the command sequence:

    $ SET DEFAULT SYS$MANAGER
    $ DIRECTORY SYS$SPECIFIC:[SYSMGR]SYSTARTUP.COM

correctly displays one copy of the file.

RESPONSE

This behavior results from the way RMS interprets search lists,
and the effects that the search lists can have when used with SET
DEFAULT.   To help explain this behavior, following is a
description of some of the underlying rules that RMS uses when
parsing file names.

    o RMS considers search-listed logical names a directive to
      "look" in n places, where n is the number of translations
      available for the logical name. It is not a wildcard. As a
      result, no effort is made to determine if any specific
      translation is unique within the search list. RMS only knows
      that it should attempt to use each translation in turn, until
      the requester is satisfied.

o SET DEFAULT sets up two file name fields, not one, and they
  are interpreted at different times. Specifically, it sets up
  SYS$DISK with the device portion of the SET DEFAULT string
  and then sets up the default directory string with the
  directory portion.   SYS$DISK is used first when parsing a
  file name and, with the advent of both concealed logical
  names and search lists, it must be evaluated whenever either
  the device or directory is missing from the filespec.

RMS is given a filespec which is missing a directory. Because of
the above rules, it first attempts to use SYS$DISK. SYS$DISK
contains a search list of two items, so RMS evaluates this logical
name twice before giving up. Note that, even though SYS$DISK will
never be used, RMS must still look through the search list in the
event that later translations will be used. Since search lists
are not wildcards, the fact that the results are not unique is
irrelevant.

We intend to document this area more clearly in a future update of
VAX/VMS.

"Does anybody really know what time it is? Does
anybody really care?"

There are several problems associated with how the time and date
are maintained in VAX/VMS. We must first present some background.

VAX/VMS makes use of several clocks, some in hardware and some in
software, to keep track of the date and time. Because none of the
available clocks solves all the problems of time keeping, they
must be used in concert and be maintained in synch by the
operating system. Under some circumstances, they might get out of
synch, causing obscure and sometimes incomprehensible problems
with the system date and time.

The master clock is maintained as a software construct by VAX/VMS.
It is a cell in the EXEC that contains the current date and time
in the VMS quadword time format. This value represents the time
elapsed since 17-Nov-1858 in tenths of microseconds.

Most VAX CPUs provide two hardware clocks from which the VMS
master clock is derived. The interval timer is used to provide an
interrupt every 10 milliseconds. At each interrupt, the quadword
master clock is incremented by the value 100,000, and time-
dependent scheduling activities are initiated.

A time-of-year (TOY) clock is built into the console subsystem of
most VAX CPUs. This clock is a 32-bit counter that is incremented
every 10 milliseconds, whether the CPU is running or not, and, if
battery backup is available, whether power is on or not. Every
time VMS is booted, the software master clock is set from the TOY
clock. This is where the trouble starts. The 32-bit, 100HZ
counter has a capacity of 497 days and, therefore, cannot be used
by itself to represent time over an indefinite period.

VMS uses the TOY clock to maintain the date and time relative to
the current year and stores the current year on the system disk in
the system image file SYS.EXE. This value is updated whenever the
system time is recalibrated (when the system is booted or when a
SET TIME command with no explicit time is entered). What is saved
in the system image is the quadword master clock value and the TOY
clock value that corresponds to it in the current year. To
recalibrate the time, the TOY clock is read and a delta is
computed from the saved TOY clock value. This delta is converted
into quadword time units and is added to the saved quadword time,
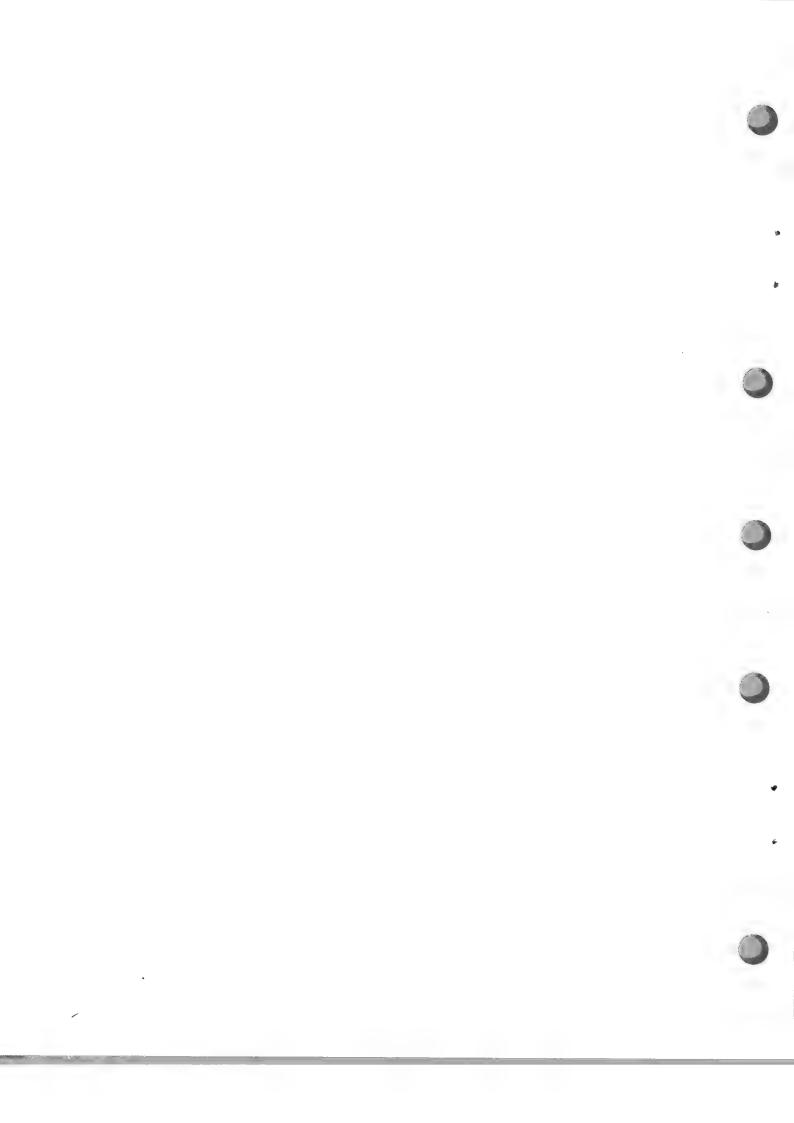
to yield the new current master clock value. If the TOY clock is found to have more than a year of time accumulated on it, one year's worth of time is subtracted and the new value is set in the TOY clock. Finally, the new TOY clock and master clock values are saved in the system image.

VMS adds a bias of 2**28 (31 days) to the time since January 1 to compute the value maintained in the TOY clock. Thus, should the TOY clock be reset or overflow, the value read will likely be less than the bias and will be rejected as an invalid clock value. Also, if the value read from the TOY clock is a day or more earlier than the saved value, it is rejected as invalid. Because of the bias, the TOY clock overflows 100 or 101 days after the first of the new year, depending on whether or not the previous year was a leap year. Thus, provided the system is rebooted or a SET TIME command is performed some time between January 1 and April 11 of each year, the TOY clock and system time will be correctly maintained indefinitely.

Problems arise when more than one copy of VMS is run on the same machine (for example, one's normal system and stand-alone BACKUP) and when new copies of the VMS EXEC are booted for the first time. For example, if two different copies of VMS are used at different times on the same machine, only one system will be presented with the opportunity to reset the TOY clock when it is first booted after January 1. The other system, when subsequently booted, will find that the TOY clock has a much smaller value than its saved value (from the last boot) and will reject the time as invalid, causing it to prompt for a new date and time.

When a new VMS system is distributed, it has assembled into it a quadword time and saved TOY clock value that represent January 1 of the current year. For example, VAX/VMS Version 4.0 was completed in October 1984; therefore, its internal time as distributed is based in 1984. Should a new copy of a system image be booted in a subsequent year, the TOY clock will be evaluated against the base date assembled into the system, and it will come up with the date set to approximately the current day in the year 1984.   This will happen, for example, with the stand-alone BACKUP kit distributed with VMS magnetic tape kits. The problem with stand-alone BACKUP is particularly bothersome because its system time is never updated when it is booted. The disk it is being booted from is either write-locked or SYS.EXE is no longer present because the first floppy or TU58 has been removed.

CUMULATIVE INDEX

Following is a cumulative listing of articles for VAX/VMS V4.n and layered products.

The following list is designed so that in future issues it can be expanded. Consequently, there are several numbers "reserved" for that purpose. Also, within each category the numbering scheme allows for expanding the primary category to include related subsets. For example, under 55.0, Utilities, 55.35 is used for the COPY utility, 55.60 is used for the DIFFERENCES utility, etc. Periodically, the components list is reviewed to insure that it accommodates the current software needs.

R = indicates a republished article

Component/ Product	Sequence Number	Title of Article	Operating System	Mon/Yr
	1.0	NEWS BULLETIN SECTION		
NEWS BULLETIN	1.1.1	IMPORTANT VAX/VMS VERSION 4.2 INFORMATION	V4.2	Sep 85
	5.0	EXECUTIVE & SYSTEM SERVICES SECTION		
SYS	5.20.1	F$GETDVI INFORMATION INVALID IF DISK NOT MOUNTED	V4.0	Jul 85
	5.20.2	EXCESSIVE MODIFIED PAGE LIST WRITING	V4.0	Jul 85
	5.20.3	GETJPI PROC_INDEX VALUE	V4.0	Jul 85
	5.20.4	SHUTDOWN WITH REBOOT_CHECK CAN FAIL	V4.0	Jul 85
	5.20.5	TODR DEFINITION REMOVED IN VAX/VMS VERSION 4.0	V4.0	Jul 85
	5.20.6	SCREEN MANAGEMENT SYMBOLS DEFINED INCORRECTLY	V4.0	Jul 85
	5.20.7	TEMPORARY MAILBOX LOGICAL NAMES	V4.0	Jul 85
	5.20.8	LACK OF DISK QUOTA CAUSES ERRFMT TO FAIL	V4.0	Sep 85
	5.20.9	GETJPI ("","TERMINAL") TRUNCATES NAMES	V4.0	Sep 85
	10.0	SYSTEM MANAGEMENT, OPERATIONS & SECURITY SECTION		
ACCOUNTING	10.5.1	PROBLEMS WITH ACCOUNTING SELECTION BY UIC	V4.0	Jul 85
	10.5.2	USER RECORD DISPLAYS SCROLL OFF SCREEN	V4.0	Jul 85
STARTUP	10.15.1	TERMINAL LOGICAL NAMES IN UVSTARTUP.COM	V4.0	Sep 85
	11.0	OPERATIONS SECTION		
LOGINOUT	11.15.1	INCORRECT VALIDATION OF MAXJOBS	V4.0	Sep 85
	11.15.2	DEFCLI PROHIBITS CLI TABLE CHANGE IN SPAWN	V4.0	Sep 85
	11.15.3	NETWORK JOBS NOT COUNTED AGAINST MAXJOBS	V4.1	Sep 85
SYSBOOT	11.30.1	TOPSYS SYSTEM ROOT IS INCORRECT	V4.0	Sep 85
SYSGEN	11.35.1	DISCREPANCY IN SCSNODE NAME LENGTH	V4.0	Jul 85
SYSINIT	11.40.1	QUOTA CACHING DISABLED ON THE SYSTEM DISK	V4.0	Jul 85
	11.40.2	SYSUAF.DAT REDEFINED FOR BYPASS AT LOGIN	V4.0	Sep 85
	12.0	SOFTWARE INSTALLATION SECTION		
UPGRADE	12.10.1	CVTUAF DOES NOT COPY USER DATA AREA	V4.0	Jul 85
	12.10.2	VMSINSTAL FAILS DURING VERSION 4.0 UPGRADE ON TU81	V4.0	Jul 85
VMSINSTAL	12.15.1	VMIBCKERR.TMP INADVERTENTLY PLACED IN SAVE SET	V4.0	Jul 85
	12.15.2	VMSINSTAL GET OPTION FAILS ON VERSION 4 UPDATE	V4.0	Sep 85

Component/ Product	Sequence Number	Title of Article	Operating System	Mon/Yr
	15.0	BATCH, PRINT, JOB CONTROLLER SECTION		
JOBCTL	15.15.1	SNDSMB WITH FILESIZ OPTION FAILS	V4.0	Jul 85
PRINT	15.25.1	SYMBIONT ISSUES BLANK PAGES WITH /SETUP	V4.0	Jul 85
	15.25.2	SUGGESTION FOR DEFAULT FORM FOR EACH QUEUE	V4.0	Jul 85
PRINT SYMBIONT	15.30.1	HOW TO PRINT HEADERS IN 80-COLUMN FORMAT	V4.0	Jul 85
	15.30.2	UNEXPECTED SYMBIONT PROCESS TERMINATION	V4.1	Jul 85
	15.30.3	CANNOT BYPASS ALL FORMATTING IN PRINT SYMBIONT	V4.0	Sep 85
	15.30.4	PRINT SYMBIONT ALLOCATES OUTPUT DEVICE	V4.0	Sep 85
	15.30.5	MULTIPLE PAGE HEADERS GENERATED BY PLOT	V4.1	Sep 85
	15.30.6	LOSS OF PRINT JOB WHEN CARRIER IS DROPPED	V4.1	Sep 85
	15.30.7	FILE LEFT OPEN BY PRINT SYMBIONT	V4.1	Sep 85
	15.30.8	IMPLICIT SPOOLING RESTRICTS USER	V4.0	Sep 85
	15.30.9	PRINT SYMBIONT PERFORMS TAB EXPANSION	V4.0	Sep 85
	15.30.10	PRINT SYMBIONT PROCESS TERMINATION	V4.1	Sep 85
	15.30.11	PRINT SYMBIONT ENTERS COMPUTE LOOP	V4.1	Sep 85
	15.30.12	MISCELLANEOUS PROBLEMS IN PRINT SYMBIONT	V4.1	Sep 85
	15.30.13	SERIAL PRINTERS ON DMF DISCONNECT	V4.0	Sep 85
	20.0	DCL SECTION		
DCL	20.5.1	CAPTIVE ACCOUNT CAUSES LOGINOUT ACCESS VIOLATION	V4.0	Sep 85
	20.5.2	CANNOT CHANGE/EXAMINE LOGICAL NAME TABLE PROT	V4.0	Sep 85
	25.0	DECnet SECTION		
DECnet	25.5.1	NETWORK JOBS USE DEFAULT DCLTABLES	V4.0	Jul 85
	25.5.2	SPURIOUS NODE UNREACHABLE ERRORS	V4.0	Jul 85
	25.5.3	STARTNET.COM INCORRECTLY PARSES NODE ADDRESS	V4.0	Jul 85
	25.5.4	STARTNET.COM FAILS TO CHECK FOR ALTPRI PRIVILEGE	V4.0	Jul 85
	25.5.5	STREAM_LF FILE TRANSFER HANGS TO NON-VMS PARTNERS	V4.0	Sep 85
	25.5.6	DECnet GIVES INCORRECT ERROR ON INVALID USER NAME	V4.1	Sep 85
	31.0	DISK & TAPE DRIVERS SECTION		
DDDRIVER	31.10.1	TU58 TIMES OUT WHEN /DATA_CHECK=WRITE IS USED	V4.0	Sep 85
	32.0	NET DRIVERS SECTION		
XDDRIVER	32.25.1	DEVICE FULL ERROR WHEN INITIALIZING DMP-11	V4.0	Jul 85
YQDRIVER	32.45.1	YQDRIVER CORRUPTS NONPAGED POOL	V4.0	Jul 85
	33.0	TERMINAL DRIVERS SECTION		
TTDRIVER	33.20.1	VT200 NOT DEFINED IN $DCDEF	V4.0	Jul 85
	33.20.2	DMA NOT SET ON DMF-32 LINES	V4.0	Jul 85
	34.0	OTHER DRIVERS SECTION		
LPDRIVER	34.20.1	SYSTEM-F-EXQUOTA ERROR ON PRINTOUT	V4.0	Jul 85
	35.0	EDITORS SECTION		
EDIT/ACL	35.5.1	EDIT/ACL DELETES ACE GRANTING ACCESS	V4.0	Sep 85
	35.5.2	PROBLEM IN REFRESH LOGIC CAUSES ACCESS VIOLATION	V4.0	Sep 85
	35.5.3	MISSING STATUS RETURN	V4.0	Sep 85

Component/ Product	Sequence Number	Title of Article	Operating System	Mon/Yr
	40.0	**FILE SYSTEMS AND RMS SECTION**		
CONVERT	40.5.1	CONVERT/RECLAIM MAY ACCESS VIOLATE	V4.0	Sep 85 R
	40.5.2	CONVERT CAN INCORRECTLY REPORT DUP AND SEQ ERRORS	V4.0	Sep 85 R
	40.5.3	CONVERT INCORRECTLY RETURNS RTL ERROR	V4.0	Sep 85
RMS	40.45.1	READ FROM SYS$OUTPUT FAILS	V4.0	Sep 85 R
	40.45.2	COPY/OVERLAY FAILS IF DESTINATION WRITE-PROTECTED	V4.0	Sep 85 R
	40.45.3	CONFUSION ON $CREATE USING SEARCH LISTS	V4.0	Sep 85 R
	40.45.4	RENAME RETURNS INCORRECT ERROR MESSAGE	V4.0	Sep 85 R
	40.45.5	ACCESS CONTROL STRING PARSED INCORRECTLY	V4.0	Sep 85 R
	40.45.6	FILE CORRUPTION WITH GLOBAL BUFFERS	V4.0	Sep 85
	40.45.7	SYS$RMSRUNDWN RETURNS INCORRECT STATUS	V4.0	Sep 85
	40.45.8	SEARCH LIST QUESTIONS	V4.0	Sep 85
	40.45.9	REMOTE COMMAND PROCEDURES FAIL	V4.0	Sep 85
	40.45.10	VERSION 4 COPY WILL NOT COPY VERSION 3 ISAM FILES	V4.1	Sep 85
	40.45.11	RMS FILE PARSE PROBLEM WITH LEVEL 8 DIRECTORIES	V4.0	Sep 85
	45.0	**RTL SECTION**		
RTL	45.1.1	VAX BASIC PROGRAMS RETURN AN INCORRECT ERL FOR ERRORS 50 AND 52	V4.0	Jul 85
	55.0	**UTILITIES SECTION**		
ANALYZE	55.5.1	ANALYZE/IMAGE REPORTS INCORRECT LINK DATE AND TIME	V4.0	Jul 85
AUTHORIZE	55.10.1	AUTHORIZE HAS TROUBLE PARSING /<ACCESS> QUALIFIERS	V4.0	Jul 85
	55.10.2	REVOKE/IDENTIFIER DOES NOT REMOVE UICS	V4.0	Jul 85
	55.10.3	CLARIFICATION OF ADD/NETWORK	V4.0	Sep 85
	55.10.4	AUTHORIZE AND DISKQUOTA DO NOT RETURN STATUS	V4.0	Sep 85
	55.10.5	PROBLEM WITH SHOW/ID FOLLOWED BY MOD/ID	V4.0	Sep 85
BACKUP	55.20.1	PROBLEM BOOTING STANDALONE BACKUP	V4.0	Jul 85
DEBUG	55.50.1	SET MODULE COMMAND TAKES TOO LONG	V4.0	Sep 85
	55.50.2	COMMA LISTS ON DEPOSIT NOT ALLOWED	V4.1	Sep 85
DIRECTORY	55.65.1	DIRECTORY OUTPUT MISSING TOTAL LINE	V4.0	Jul 85
	55.65.2	DIRECTORY MAY DISPLAY NONEXISTENT FILES	V4.0	Jul 85
INITIALIZE	56.5.1	INITIALIZE/INDEX:BLOCK=N NOT RECOGNIZED	V4.0	Jul 85
INSTALL	56.10.1	INABILITY TO INSTALL EXECUTABLE IMAGES	V4.0	Jul 85
LIBRARIAN	56.15.1	PROBLEM DECOMPRESSING A LIBRARY	V4.0	Jul 85
LINKER	56.20.1	LINKER OPEN FILE LIMIT PROBLEM	V4.0	Jul 85
	56.20.2	LINKER REJECTS VALID FILE NAMES IN OPTIONS FILES	V4.0	Jul 85
	56.20.3	VERSION 4.0 IMAGES LARGER THAN VERSION 3.0 IMAGES	V4.0	Sep 85
MAIL	56.30.1	MAIL CANNOT RUN ON A GIGI TERMINAL	V4.0	Sep 85
MONITOR	56.40.1	FOREIGN TERMINAL SUPPORT DOES NOT WORK	V4.0	Sep 85
	56.40.2	MONITOR'S VIRTUAL MEMORY USAGE GROWS CONTINUOUSLY	V4.0	Sep 85
PURGE	56.52.1	PURGE CAN INCORRECTLY DELETE FILES	V4.0	Sep 85
SHOW	56.75.1	RANDOM BROADCAST CLASSES DISABLED	V4.0	Jul 85
SET	56.80.1	SET PASSWORD SIGNALS ERRORS TWICE	V4.0	Jul 85
	56.80.2	VOLUME RETENTION DATES OVERRIDE SET FILE DATES	V4.0	Sep 85
	56.80.3	SET PASSWORD ALWAYS RETURNS SUCCESS STATUS	V4.1	Sep 85

COMPONENTS LIST

DISPATCH INDEX

32.35	XGDRIVER
32.40	XMDRIVER
32.45	XQDRIVER
32.50	XWDRIVER

33.0   Terminal Drivers

33.5	CTDRIVER
33.10	DZDRIVER
33.15	RTTDRIVER
33.20	TTDRIVER
33.25	YCDRIVER
33.30	YFDRIVER

34.0   Other

34.5	CRDRIVER
34.10	DXDRIVER
34.15	LADRIVER
34.20	LPDRIVER
34.25	LTDRIVER
34.30	MBDRIVER
34.35	MBXDRIVER
34.40	PADRIVER
34.45	PUDRIVER
34.50	XADRIVER
34.55	XGDRIVER
34.60	XIDRIVER
34.65	XJDRIVER
34.70	XKDRIVER
34.75	XMDRIVER

144

SOFTWARE PROBLEMS OR ENHANCEMENTS

Questions, problems, and enhancements to DIGITAL software should be reported on a Software Performance Report (SPR) form and mailed to the SPR Center at one of the following Digital Offices: *(SPR forms are available from the SPR Center).*

Areas Covered	SPR Center
United States; remainder of Far East, Middle East, Africa Latin America	Corporate Administrative Services Group P.O. Box F Maynard, MA 01754
Canada	Digital Equipment of Canada, Ltd. P.O. Box 13000 Kanata, Ontario Canada, K2K 2A6
United Kingdom, Bahrein, Egypt, Iraq, Jordan, Kuwait, Lebanon, Libya, Qatar, Oman, Saudi Arabia, Syria, United Arab Emirates, Yemen, Arab Republic	Digital Equipment Co. Ltd. 2 Cheapside GB - Reading, Berkshire RG1 7AA England
Australia, New Zealand	Digital Equipment Aust. Pty. Ltd. P.O. Box 384 Chatswood, New South Wales 2067 Australia
Brazil	Digital Equipment Comercio e Industria Ltda. Avenida Augusto Severo, 156-A 20021 Rio de Janeiro, RJ Brazil
Caribbean	Digital Equipment Latin America P.O. Box 11038 Fernandez Juncos Station Santurce 00910 Puerto Rico
France	Digital Equipment France Cidex L225 18 Rue Saarinen F-94528, Rungis France
Italy	Digital Equipment S.p.A. Viale Fulvio Testi, 11 Ang. Via Gorki 105 I-20092 Cinisello Balsamo Milan Italy
Japan	Digital Equipment Corp. Intl. Japan Sunshine 60, P.O. Box 1135 1-1 Higashi Ikebukuro 3-Chome, Toshima-Ku, Tokyo, 170 Japan
Belgium, Holland, Luxemburg	Digital Equipment B.V. Kaap Hoorndreef 38 NL-3563 AV Utrecht Holland

145

Sweden	Digital Equipment AB P.O. Box 1250 S-17124 Solna 1 Sweden
Denmark	Digital Equipment Corp. AS Kristineberg 3 DK-2100 Copenhagen 0 Denmark
Finland	Digital Equipment Corp. Oy PL 16 SF-02201, Espoo 20 Finland
Norway	Digital Equipment Corp. A/S Pottemakerveien 8 N-Oslo 5 Norway
Austria, East Germany, West Germany, Poland, Hungary, Rumania, Czechoslovkia, Russia, Bulgaria	Digital Equipment Corp. GmbH Rheinstrasse 28 D - 8000 Munich 40 West Germany
Israel	Decsys, Computers Ltd. 4, Yirmiyahu Str. IL-63505 Tel Aviv Israel
Greece, Portugal, Spain, Switzerland, Yugoslavia, (Morocco, Algeria, Tunisia, Cyprus, Turkey, Malta)	Digital Equipment Corp. SA 9, Route des Jeunes Case Postale 191 CH-1211 Geneva 26 Switzerland
Mexico	Digital Equipment de Mexico, S.A. de C.V. Ave. Lopez Mateos 427, 1st. Floor Guadalajara Jalisco Mexico
China	Digital Computer Hong Kong Ltd. 1303-1309 Dominian Ctr. 43-59 Queen's Road East Wanchai Hong Kong

This data sheet explains what software licenses are and why customers must obtain a software license to run any item of DIGITAL proprietary software.

DIGITAL does not sell software; DIGITAL offers software under a license agreement. DIGITAL has a license agreement for source software and object software. Since DIGITAL software programs are made available primarily in object code, this data sheet focuses on the purchase of object programs.

Introduction to Software Licensing

When DIGITAL hardware is purchased, all rights of ownership (legally called "title") to the hardware pass to the customer. This is not the case with software. DIGITAL regards software as proprietary information. Since software is easily reproduced, it must be legally protected from improper copying. Therefore, DIGITAL uses a combination of trade secret and copyright legal protection for software. DIGITAL protects its investment by retaining title to its software at all times and requires anyone wishing to use it to obtain a license.

How DIGITAL Licenses Software

The license agreement for object programs is contained in DIGITAL's standard Terms and Conditions of Sale and Corporate Volume Purchase Agreements instead of as a separate agreement. Therefore, when software products are purchased under DIGITAL's Terms of Sale, the software license agreement is made at the same time.

Key Principles of the License Agreement

Object code is licensed for single use. This means obtaining a license for a product allows the associated software to be used on the "single" CPU on which it was first installed. Other key points are as follows:

- If the licensed CPU temporarily malfunctions, the software may be run on another machine while the CPU is down.

- Copies of the software may be made for backup purposes if appropriate proprietary and copyright notices are included.

- The software may be modified or merged with other software if appropriate proprietary and copyright notices are included.

- The software may be used by the customer's employees and its agents directly concerned with the internal use, but may not be made available to anyone else.

Modification to the Software Product

Any modification to licensed software does not exempt the product from DIGITAL license terms. Every line of code from a software product falls under the terms of the license. Only those modifications that are not part of the original software are the customer's property. It is important to note that warranty on the product is limited to the original software supplied by DIGITAL.

Transferability of Licensed Software Products

License Transfer - A license agreement does not automatically allow transfer of licensed software to another party or another CPU. If the customer intends to sell the licensed CPU and pass on the software with the sale or move the software onto another CPU, permission must be obtained from DIGITAL. A case-by-case License Transfer is required to relicense the software.

Software Sublicensing - DIGITAL customers with a purchase agreement authorizing sublicensing, such as OEMs, may transfer licensed object products to their customers without a License Transfer. A valid sublicense, executed by an OEM with its customer, gives the OEM's customer the same license rights and responsibilities as a license agreement made directly with DIGITAL.

Source Software

Sources are only available for selected products. A license agreement for source software must be separately executed for each facility/location which intends to purchase sources in machine-readable, listing, or microfiche form. Further information and availability of sources can be found in the applicable Software Product Description (SPD).

Software Warranty

Each licensed software product offered has an SPD describing the warranty commitment for the product. Software products under DIGITAL warranty must conform to the description provided for a 90-day period, which generally begins upon product installation or 30 days after delivery. All other products are provided AS IS, without warranty. The SPD clearly states under which warranty category the product falls.

Purchasing the License for the Software Product

A license must be obtained for each CPU on which the licensed software will be used (unless otherwise specified by DIGITAL).

A Single-use License for object code is generally ordered according to the type/classifcation of the CPU or system configuration intended to run the product. Further information and availability can be found in the applicable SPD.

Software Product

A license is a prerequisite to purchase the associated software. The Media and Documentation Option for a product is ordered according to media type. Further information and availability of media can be found in the applicable SPD.

Purchasing Software Product Revisions/Updated Versions

If a licensed customer is not covered by a product service agreement, updated versions can be purchased when they are made generally available. Updated versions are ordered according to media type. A customer can also choose to run updated versions on additional CPUs, but not purchase multiple media distributions. If this is the case, the Software Revision Right-to-Copy option must be purchased for each CPU which runs the updated version.

Software Product Services

A licensed customer can purchase annual product service agreements to receive updated versions on media when available. A customer may choose to copy updated versions onto additional CPUs during this service agreement period. In this case, the software Service Right-to-Copy must be purchased for each CPU which runs the updated version. Further information and availability can be found in the applicable SPD. Your local DIGITAL office can be contacted for additional assistance.

148

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY

BENEFITS OF BELONGING

The Digital Equipment Computer Users Society (DECUS) is one of the largest and most respected users groups in the computer industry today. Membership in DECUS, which is free and voluntary, provides the individual user with information and services not found anywhere else.

DECUS provides an environment where users of Digital Equipment Corporation products can share information with other users and with DIGITAL. Members can find out the latest news on DIGITAL's hardware, software, and educational products. The feedback exchange with DIGITAL allows the users of DIGITAL's products to have a voice in the company's future.

Founded in 1961, DECUS now has three autonomous areas worldwide - DECUS U.S., DECUS Europe, made up of eight independent chapters, and DECUS GIA (General International Area), made up of four independent chapters. DECUS services and activities are shared between these chapters through mutual agreements.

All DECUS services promote the exchange of information in a noncommercial environment. Included in these services are:

Special Interest Groups (SIGs)

These groups, formed around an area of common interest, exist for a variety of hardware, operating systems, languages, applications, and marketing areas. Participation in these groups allows fellow users to exchange information and share technical expertise in the areas of most interest to the users.

Local Users Groups (LUGs) and National Users Groups (NUGs)

LUGs and NUGs are licensed groups of individuals who gather to share information with other users on a periodic basis. Not only do they have common professional interest, but they also have geographic and cultural ties. DIGITAL representatives attending these meetings often unveil new products and services and supply updates on existing policies and procedures.

Symposia

DECUS holds symposia each year in the different chapters, two per year in the U.S. These meetings provide a unique opportunity for users with a wide spectrum of experience to meet for up to five days of intensive technical exchange. Symposium activities include workshops, clinics, panels, tutorials, and formal paper presentations. DIGITAL participates in symposia by sending Product Group managers and developers to discuss strategies, products, problems, and solutions.

Publications

The flow of information among users, as well as between users and DIGITAL, is the primary goal of DECUS. Various publications generated by DECUS support this communication. They include chapter newsletters and *The Proceedings*, a technical volume published after each symposium. DECUS also publishes Special Interest Groups' newsletters that provide information pertaining to specific DIGITAL products.

Program Library

The DECUS Program Library is the main vehicle for the exchange of software among users of all DIGITAL systems. The Library contains over 1000 software programs written and voluntarily submitted by users. These programs include compilers, editors, utilities, numerical and statistical functions, as well as games and graphic routines. The Library publishes an annual software catalog that lists and describes all the DECUS programs available to all users for a minimal charge.

You are cordially invited to join over 60,000 other users of DIGITAL products around the world and begin to share your experiences, both successes and problems.

For more information, contact the appropriate DECUS chapter office listed here.

DECUS CHAPTER OFFICES — WORLDWIDE

DECUS U.S.

DECUS, U.S. Chapter
249 Northboro Road (BPO2)
Marlborough, Massachusetts 01752
U.S. Activities: (617) 480-3259 (3302)
Library: (617) 480-3521
Finance and Administration: (617) 480-3634

DECUS Europe

DECUS At Large (in Europe)
C.P. 510
CH-1213 PETIT-LANCY 1/GE
Switzerland

DECUS France
B.P. 136
F-91004 EVRY CEDEX

DECUS Holland
Kaap Hoorndreef 38
NL-3563 AV UTRECHT
The Netherlands

DECUS Muenchen
Freischuetzstrasse 91
D-8000 MUENCHEN 81
Federal Republic of Germany

DECUS Italia
Viale Fulvio Testi 11
I-20092 CINISELLO BALSAMO

DECUS Nordic
S-172 89 SUNDBYBERG
Sweden

DECUS U.K., Ireland and
Middle East
P.O. Box 53
READING, RG2 OTW
U.K.

DECUS Switzerland
Schaffhauserstrasse 144
8302 Kloten
Switzerland

DECUS GIA (General International Area)

DECUS Australia
Northern Tower, Chatswood Plaza
Railway Street
Chatswood, New South Wales 2067
Australia
Phone: (02) 412.5237

DECUS Canada
100 Herzberg Road
P.O. Box 13000
Kanata, Ontario K2K 2A6
Canada
Phone: (613) 592-5111, ext. 2115

DECUS Japan
Nihon Digital Equipment KK
Sunshine 60, P.O. Box 1135
1-1, Higashi Ikeburo 3-Chome
Toshima-ku, Tokyo 170
Japan
Phone: [81]-(3)-9897111

DECUS GIC
100 Nagog Park AKO1-1/B11
Acton, Massachusetts 01720
U.S.A.
Phone: (617) 264-6561

July 1984

DECUS